BMS/TS 8
Administrator's Guide

GT Software, Inc.
1314 Spring Street NW
Atlanta, Georgia USA 30309
404–253–1300
<www.gtsoftware.com>

Copyright and Legal Notices

GT Software, Inc.
1314 Spring Street NW
Atlanta, Georgia USA 30309
Phone: 404–253–1300
Fax: 404–253–1314
E–mail: sales@gtsoftware.com or support@gtsoftware.com
URL: http://www.gtsoftware.com/

Notice

This PDF version of the BMS/TS product documentation was produced using HTMLDOC <http://www.easysw.com/htmldoc/>, which is available under the GNU General Public License <http://www.fsf.org/copyleft/gpl.html>.

The complete BMS/TS documentation is supplied with the product in HTML format. This PDF is provided for the convenience of those customers who prefer hardcopy documentation.

BMS/TS Doc. Home

BMS/TS Administrator's Guide Home

Copyright and Legal Notices

Top of page

BMS/TS Doc. Home

Contents:

Introduction to BMS/TS

BMS/TS is a software tool for designing BMS maps for CICS applications. It improves programmer productivity, increases application efficiency, and creates user–friendly help displays. But BMS/TS is more than a map generator. BMS/TS consists of five primary components that work together to let the programmer rapidly develop a program's entire user interface. Those components are:

- Map Generator,
- Input Rules,
- Table Manager,
- Help Screens,
- Batch Programs.

Note: When using BMS/TS, you will not interact with these components as separate applications. Keep in mind, though, that some of these components may be licensed separately, and therefore may not be available.

Map Generator

The BMS/TS Map Generator lets you create CICS/BMS maps quickly and easily. With the Map Generator, you create maps on your terminal screen. Then, using the online submit function, you catalog the maps to the system library –– all without ever leaving BMS/TS or having to assemble any code. When complete, you have a working, perfectly aligned physical and symbolic map.

To help in the creation process, the Map Generator includes many time–saving features. With these features, you can:

- Define extended attributes, including highlighting, color, and validation,
- Automatically add prefixes to field names,
- Define group fields,
- Create field, column, line, and user–defined arrays,
- Copy and move blocks of fields,
- Display a list of a map's named fields, along with their lengths and attributes.

Refer to Working with Maps for instructions on generating maps.

Input Rules

Input rules let you add a validation process to a map, instructing BMS/TS to evaluate and act on data entered in the map. Input rules go far beyond the simple extended validation provided by the extended attribute of a standard BMS map. For example, input rules function independently from the terminal device, so they will work regardless of the terminal device type.

You can create input rules to perform a variety of tasks on the data entered in a field on the map, such as:

- Force a field to be entered,
- Force a field to be filled,
- Format dates,
- Initialize empty fields,
- Perform range checks,
- Force field values,
- Perform table look–ups,
- Return custom error messages.

You write input rules using a special Input Rule Editor. The structure of the Input Rule Editor makes it simple to write these input validation and editing routines that you would otherwise have to code in the application program.

Refer to Working with Input Rules for instructions on creating input rules.

Table Manager

The BMS/TS Table Manager enables you to create and maintain validation and look–up tables externally from your program. Without the Table Manager, tables would typically be embedded in program code, which when changes are required, would force programmers to go into the source code to update tables and then recompile their programs.

Using the Table Manager, you can refer to validation and lookup tables with your input rules. For example, your input rules can compare the operator's input to a list of valid entries stored in a table. If the input is valid, the input rules can return a cross–referenced value from the table to either the terminal or to the application. If the input is not valid, the input rules can return a custom error message.

Refer to Working with Tables for instructions.

Help Screens

Help screens let you add customized information displays to applications. You create help screens as you would any other map. You can implement these help screens easily without changing the application program.

Refer to Working with Help Screens for instructions on creating help screens.

Batch Programs

BMS/TS also includes batch programs to help you manage how BMS/TS functions, as well as the maps and tables you create with BMS/TS. Some of the operations you can perform with batch programs include:

- Back up and restore BMS/TS libraries,
- Print maps or a list of maps,
- Copy, rename, or erase maps and tables within a library,
- Convert physical and symbolic BMS maps for revision with BMS/TS,
- Generate BMS macro source code from BMS/TS maps.

Refer to Appendix B for more information about BMS/TS batch functions.

Transaction Monitor Table

The BMS/TS Transaction Monitor works with your input rules and/or help screens to enhance the applications. By adding entries to the Transaction Monitor table, you indicate which maps use input rules or help screens. The Transaction Monitor then monitors the data, commands, and requests sent from the terminal and directs them to the appropriate BMS/TS component.

> For example, after you provide the information to the Transaction Monitor, pressing a function key can automatically display a corresponding help screen. Or, the Transaction Monitor can activate the appropriate validation routine (based on your definitions) for all maps for which you have created input rules.

Understanding How BMS/TS Works

Here's how the BMS/TS components interact within your application.

1. Users work with a map in your application.

2. Users enter data, commands, or press function keys to make requests.

3. The Transaction Monitor examines all requests: help requests, entered data, and commands. It passes help requests to the help screens component, and everything else to the input rules.

4. The Transaction Monitor processes help requests and displays the corresponding help screen. The Transaction Monitor processes all other data, commands, and requests through the input rules.

5. Depending on the request, BMS/TS performs one of the following functions:

   - Displays the appropriate help screen,
   - Reads and validates the information in the data stream against the input rules, responding to any commands, requests or invalid data as you have designated. When all entered data is valid, the input rules pass the data to your application program.

Top of page

Contents:

Related topics:

- Change Map Option Defaults
- Change Submit Option Defaults
- Edit JCL
- Update Security
- Change Product Options
- User Exits

Administrator's Guide Home
BMS/TS Doc. Home

Customizing BMS/TS

You can customize BMS/TS to conform to the needs of your company. The process of customizing BMS/TS involves altering the default records supplied with BMS/TS or creating default records for individual operators. You can specify:

- The programming language you use for your maps,
- The hardware your operators are using,
- How you want your programmers to interact with BMS/TS when creating maps,
- Your specifications for submitting maps to the CICS system library for cataloging,
- Your JCL conventions,
- How you want to restrict programmer access to certain functions,
- Your standards for multiple libraries,
- Your requirements for security switches, dates, editing patterns, expanded function keys, and passwords,
- How you use other GT products and the Transaction Monitor.

Understanding BMS/TS Default Records

BMS/TS supplies the following default records:

- Default Map Definitions
  The Map Definition Default Record controls a variety of global definitions for your maps, including such features as the programming language you use for your maps, the hardware your operators are using, and how you want your programmers to interact with BMS/TS when creating maps.

- Default Submit Options
  The Submit Record controls the settings for submitting maps to the system library.

- Default JCL Member
  The JCL Member controls the JCL conventions your company uses to submit maps and tables to be catalogued in the CICS system library

- Default *ADMIN Record
  The *ADMIN record lets you control BMS/TS in the following ways:
  - ♦ Managing multiple libraries,
  - ♦ Controlling security switches, dates, and editing patterns,
  - ♦ Supporting expanded function keys,
  - ♦ Using a master password,
  - ♦ Using other GT products and the Transaction Monitor.

You can also create your own Default Security Record.

> **Note:** Because a security record contains specific information that is unique for your company, BMS/TS does not provide a default security record. Instead, you can create a default security record that controls the authorization of your programmers (both individual and system wide) to access certain system functions.

Customization Options

You have the following options for customizing BMS/TS:

Change Map Option Defaults	Modify the defaults (globally or for an individual operator) for various map settings, including the programming language you use, the types of terminals in your installation, etc.
Change Submit Option Defaults	Modify your default Submit record (globally or for an individual operator) by altering your settings for submitting maps to the CICS system library.
Edit JCL	Modify the default JCL member SUBMIT (provided by BMS/TS), as well as create other JCL members using the JCL conventions of your company.
Update Security	Create a security record to protect access to all desired BMS/TS functions, files, and applications.
Change Product Options	Modify BMS/TS so that it conforms to the following: • Your standards for multiple libraries, • Your requirements for security switches, dates, editing patterns, expanded function keys, and passwords, • How you use other GT products and the Transaction Monitor.
User Exits	Write user exits so that you can branch from BMS/TS processing, perform other routines, then return to BMS/TS.

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Customizing BMS/TS

Administrator's Guide Home
BMS/TS Doc. Home

Change Map Option Defaults

You can update your default map settings. You can also use this function to create a default record for an individual operator.

The Map Definition options allow you to specify the programming language you are using, the user's terminal type, and other characteristics of your maps.

> **Note:** If you want to create a map option defaults record for an individual operator, you must require operators to sign on to BMS/TS using an operator ID. In order for BMS/TS to locate the correct default record, you must also use the same "name" for both the map option defaults record and for the ID: for example, if a developer's ID is DEV01, you would also name the map option defaults record DEV01.

Identify the Operator ID

Entering an operator ID lets you indicate whether you want to change the map option defaults globally, or for an individual operator. BMS/TS activates these options whenever the appropriate operator(s) begins creating a map.

1. From the Main Menu, select the Product option and press Enter. The Product Menu displays (view sample).

2. From the Product Menu, select the Change Map Option Defaults option and press Enter. The Enter Operator ID screen displays (view sample). The cursor moves to the Operator ID field.

3. Perform the desired action:

If you want to...	Then...
Change the existing map option defaults record for all operators	Type DEFAULT.
Create or change the default for a single operator	Type the desired operator's sign–on ID.
Display a list of map option default records in the current BMS/TS library	Press F2.

4. If the map option defaults record has been password–protected, press Tab to move to the Password field and type the member password.

   Note: You can also use this field to add a password to an unprotected record. Remember that if

you assign a password to a defaults record, you must enter the password each time you edit the record with BMS/TS.

5. Press Enter. BMS/TS displays the Map Definition Options screen (view sample).

Define Map Option Defaults

Defining map options lets you modify the defaults (globally or for an individual operator) for various map settings.

After you identify the Operator ID, the Map Definition Options screen appears (view sample).

1. Press Tab to move to the entry fields you want to modify, then type over the currently displayed value.

   Note: Press F1 from any field on the screen to display online help. Keep in mind that options for Language, Upper and Lower Case, Sound Alarm, Resets, Extended Attributes, and Page Build normally stay constant at a company, project, or application level.

2. Repeat step 1 until you make all of your modifications.

3. Press F3. BMS/TS saves the new default record and returns to the Enter Operator ID screen.

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Customizing BMS/TS

Administrator's Guide Home
BMS/TS Doc. Home

Change Submit Option Defaults

You can update your default submit settings. You can also use this function to create a default submit record for an individual operator.

The Submit Option defaults allow you to select options for cataloging and submitting maps to the CICS system library. Keep in mind that when submitting a map, BMS/TS creates the physical map and the symbolic map.

> **Note:** If you want to create a submit option defaults record for an individual operator, you must require operators to sign on to BMS/TS using an operator ID. In order for BMS/TS to locate the correct default record, you must also use the same "name" for both the submit option defaults record and for the ID. For example, if a user's ID is TECH01, you would also name the submit option defaults record TECH01.

Identify an Operator ID

Selecting an operator ID lets you indicate whether you want to change the submit option defaults globally, or for an individual operator. BMS/TS activates these options whenever the appropriate operator(s) begin submitting a job.

1. From the Main Menu, select the Product option and press Enter. The Product Menu displays (view sample).

2. From the Product Menu, select the Change Submit Option Defaults option and press Enter. The Select an Operator ID screen displays (view sample). The cursor moves to the Member name field.

3. Perform the desired action:

If you want to...	Then...
Change the existing submit option defaults record for all operators	Type DEFAULT.
Create or change the default for a single operator	Type the desired operator's sign–on ID.
Display a list of submit option default records in the current BMS/TS library	Press F2.

Note: Creating an individual operator's default submit record produces a library member using a set name of BMS/TS and a member name of the operator's ID. For example, if a developer's ID is DEV01, you would create a defaults record with a library name of BMS/TS and a member name of DEV01.

4. If the submit option defaults record is password–protected, press Tab to move to the Password field and type the password.

   Note: You can also use this field to add a password to an unprotected record. Remember that if you assign a password to a defaults record, you must enter the password each time you edit the record.

5. Press Enter. BMS/TS displays the Batch Job Submission screen (view sample).

Define Submit Option Defaults

Defining submit options lets you modify the defaults (globally or for an individual operator) for submitting a job to your CICS system library.

After you identify an operator ID, the Batch Job Submission screen appears.

Note: The screen you see depends on your operating system (view sample OS/390 screen).

1. Press Tab to move to the fields you want to modify, then type over the currently displayed value. Press F1 from any field on the screen to display the purpose, valid entries, and examples for the field.

   Keep in mind that these settings are not absolute. Your programmers can still alter the submit options for an individual set and/or member during the submission of the set/member.

   Note: If desired, you can also customize the BMS/TS default field labels to labels that meet your company's standards. For example, if you change the @PVSNAME field, BMS/TS passes the job name you enter here to your JCL wherever it finds the variable @PVSNAME.

2. Press F3. BMS/TS saves the new default record and returns to the Select an Operator ID screen.

The following table describes each of the customizable field labels that appear on this screen.

Field Label	Description
@PVSNAME	POWER/VSE job name
@JOBNAME	OS/390 job name
@JCLASS	Job class
@JDISP	POWER/VSE job disposition
@PCLASS	PRINT or LIST class
@VSENAME	VSE job name
@USER001–002	User–defined value. BMS/TS inserts the up to 8 character value that you

	enter for each of these field labels whenever the @USER001 and/or @USER002 variables appear in the JCL.
@USER003–004	User–defined value. BMS/TS inserts the up to 40 character value that you enter for each of these field labels whenever the @USER003 and/or @USER004 variables appear in the JCL.

Note: Remember, your programmers must use the variables in this table in the actual JCL so that BMS/TS can pass the correct value to the JCL.

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Edit JCL

You can create or update JCL members using the BMS/TS freestyle editing screen to key in your JCL commands.

> **Note:** BMS/TS provides a default JCL member called SUBMIT. You can customize this default JCL member to your company's standards, or you can create individual JCL members for individual operators. You can name individual JCL members in any way you wish.

Select a JCL Member

Selecting a JCL member lets you indicate whether you want to edit an existing JCL member, or create a new JCL member.

1. From the Main Menu, select the Product option and press Enter to display the Product Menu (view sample).
2. From the Product Menu, select the Edit JCL option and press Enter. The Select a JCL Member screen displays (view sample). The cursor moves to the JCL name field.
3. Type the name (1–8 characters) of the JCL member you want to edit or create.

   Note: To display a list of JCL members in the current BMS/TS library, press F2.
4. If the JCL member is password–protected, press Tab to move to the Password field and type the member password.

   Note: You can also use this field to add a password to an unprotected member. Remember that if you assign a password to a JCL member, you must enter the password each time you edit the JCL member with BMS/TS.
5. Press Enter. BMS/TS displays the Job Control Language Editor screen.

Edit a JCL Member

After you select a JCL member, the Job Control Language Editor screen appears.

> **Note:** The screen you see depends on your operating system (view sample OS/390 screen).

See Appendix A, Using the BMS/TS Line Editor, for information about the line commands (LC), command line entries, and function keys that appear on this screen.

1. Press Tab to move to the Desc field. If you want a comment displayed in the library listing, type a new description for your JCL member, or edit the existing description. You can enter up to 35 characters.

2. Press Tab to move to the first line you want to add or edit.

3. Type your JCL statements. If desired, you can also include three different types of variables in your JCL:

User−defined variables	Variables you define to identify specific information related to a submit job. For example, you can set up a variable to identify a job name or to pass a specific value from the actual data entered to the JCL.
System variables	Variables with preset definitions that allow BMS/TS to include actual system values (such as the system date or time) in the JCL.
BMS/TS variables	Variables with preset definitions that allow you to include BMS/TS source or object code in the JCL. For example, you can use one of these variables to include a physical map in object code in the JCL.

4. Press F3. BMS/TS saves the new JCL member and returns to the Select a JCL Member screen.

Variable Names

Valid variable names for these three types appear in the following tables.

> **Note:** BMS/TS examines the JCL card for any of these variable names before passing each card to the spooler. The value entered on the submit screen replaces the variable name in the card and BMS/TS shifts the card to compensate for any size difference.

User−defined Variables

Variable Name	Description
@PVSNAME	POWER/VSE job name
@JOBNAME	OS/390 job name
@JCLASS	Job class
@JDISP	POWER/VSE job disposition
@PCLASS	PRINT or LIST class
@VSENAME	VSE job name
@USER001−002	Value you define for use anywhere in the JCL to be replaced with an up to 8 character value that the programmer enters on the submit screen.
@USER003−004	Value you define for use anywhere in the JCL to be replaced with an up to 40 character value that the programmer enters on the submit screen.

System Variables

Variable Name	Description
DATE	Current system date in *mm*/*dd*/*yy* format.
TIME	Current system time in *hh*:*mm*:*ss* format.
TERM	Identification number of terminal submitting the JCL.
OPID	Identification number (if any) of operator submitting the JCL.
MAPSET	Mapset name entered when initiating the submit job.
MAPNAME	Map name entered when initiating the submit job.
TABNAME	Table name entered when initiating the submit job.

BMS/TS Variables

Variable Name	Description
MAP	BMS/TS replaces this variable with the object code of the physical map you are submitting.
DSECT	BMS/TS replaces this variable with the source code of the symbolic map you are submitting.
TABLE	BMS/TS replaces this variable with the object code of the table you are submitting.

Note: When you use any of the BMS/TS variable names, make sure that the variables appear as the first data on the JCL card image.

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Understanding BMS/TS Security
- Customizing BMS/TS

Administrator's Guide Home
BMS/TS Doc. Home

Update Security

You can customize the BMS/TS security system to restrict any BMS/TS area or function from unauthorized access. Read Understanding BMS/TS Security for more detailed information about security options and for specific examples.

Updating security enables you to restrict access (globally or for an individual operator) to individual mapsets, special sets, and/or any BMS/TS function or option.

> **Note:** If you want to create a security default record for an individual operator, you must require operators to sign on to BMS/TS using an operator ID. In order for BMS/TS to locate the correct default record, you must also use the same "name" for both the security default record and for the ID. For example, if a user's ID is TECH01, you would also name the security default record TECH01.

Select an Operator ID

Selecting an operator ID lets you indicate whether you want to create or update the security settings for all operators, or for an individual operator.

1. From the Main Menu, select the Product option and press Enter. The Product Menu displays (view sample).
2. From the Product Menu, select the Update Security option and press Enter. The Edit Operator Security screen displays (view sample). The cursor moves to the Operator Identification field.
3. Perform the desired action:

If you want to...	Then...
Create or change the security record for all operators	Type DEFAULT.
Create or change the security record for an individual operator	Type the desired operator's sign–on ID (1–8 characters).
Display a list of security records in the current BMS/TS library	Press F2.

4. If the security record is password–protected, press Tab to move to the Password field and type the password.

Note: You can also use this field to add a password to an unprotected record. Remember that if you assign a password to a security record, you must enter the password each time you edit the record.

5. Press Enter. BMS/TS displays the Operator Security Editor screen (view sample).

Edit Operator Security

Editing operator security lets you enter the areas, options, functions, and/or applications that you want to protect from unauthorized entry.

After you complete the Edit Operator Security screen, the Operator Security Editor screen appears (view sample).

Note: See Appendix A, Using the BMS/TS Line Editor, for information about the line commands (LC), command line entries, and function keys that appear on this screen.

1. Press Tab to move to the Desc field. If you want a comment displayed in the library listing, type a new description for your security record, or edit the existing description. You can enter up to 35 characters.

2. Press Tab to move to the first A/R field.

3. The A/R field lets you indicate whether you want to restrict or allow access to a specific BMS/TS area, option, function, and/or application. If you want to allow access, type A (Allow). If you want to restrict access, type R (Restrict).

4. The cursor moves to the Tran field. Use this field to indicate the specific transaction code to which BMS/TS should apply this line of security. You can use asterisks (*) as wild cards, if necessary.

   For example, if you enter GT* in this field, BMS/TS applies this line of security to all transaction codes beginning with GT. This is especially helpful if you can use more than one transaction to start BMS/TS.

5. Press Tab to move to the Command field.

6. Type the command code for the BMS/TS feature from which you want to restrict user access.

7. Press Tab to move to the Type field.

8. Type the name of the mapset or special set you want to protect.

9. Press Tab to move to the Set Name field.

10. If you want to protect a specific map or member, type the member name. Otherwise, skip this field.

11. Press Tab to move to the Comments field.

12. Type any comments you wish to record about this line of security options.

13. Repeat steps 1–12 to define all desired lines of security.

14. Press F3. BMS/TS saves the new or edited security record and returns to the Edit Operator Security screen.

Command Codes

Command	Description
ADD	Creates a set and/or member.
COPY	Duplicates a set and/or member.
DELETE	Removes a set and/or member.
DISPLAY	Shows a set and/or member.
DLIST	Lists the contents of a directory.
FETCH	Retrieves a BMS map from one library into another.
LINK	Opens a region of another library without leaving the current library.
MERGE	Retrieves a set and/or member into BMS/TS.
PASSWORD	Changes a master password.
RESET	Changes an update flag on a member.
RENAME	Changes the name of a set and/or member.
UPDATE	Edits a member.

The following command codes are only valid if you enter *SYSTEM in the Type field:

Command	Description
ON	Loads the Transaction Monitor. This activates the BMS/TS help screens and input rules.
NEWCOPY	Refreshes the Transaction Monitor.
MONOFF	Temporarily suspends the Transaction Monitor.
DELOAD	Deactivates all monitoring programs and releases all work areas acquired by BMS/TS.
TLIST	Tests any table you have created for use with input rules.
TSTATUS	Displays the status of tables loaded into memory for processing but not yet released.

You should only use the following command codes under the supervision and request of GT Software Technical Support. Please contact Technical Support for additional information.

Command	Description

TRACE	Activates/deactivates the BMS/TS help user trace facility.
MDEBUG	Activates/deactivates the BMS/TS help error identification facility.
DUMPGT	Display any CICS storage areas contained within the CICS system.

You should only use the following command codes if the BMS/TS 3270 data compression facility has been licensed for use at your company:

Command	Description
COMPON	Activates the 3270 data compression facility.
COMPOFF	Deactivates the 3270 data compression facility.
CSTATUS	Displays the accumulated 3270 data stream compression statistics.

Special Set Names

Special Set	Description
***ADMIN**	Administration records created when you installed BMS/TS.
***DEMO**	Working demonstrations you constructed with BMS/TS.
***JCL**	Job Control Language that you write and store in GT libraries and use to submit your maps and tables.
***MAPS**	Mapsets and maps you create with BMS/TS. **Note:** If you do not specify a special set name, BMS/TS defaults to *MAPS.
***OPID**	Security records for individual users of BMS/TS.
***SYSTEM**	Transaction Monitor records created internally as you update the Transaction Monitor Table.
***TABLE**	Tables of data you create to enhance your input rules.

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Update Security

Administrator's Guide Home
BMS/TS Doc. Home

Understanding BMS/TS Security

The BMS/TS internal security option controls individual user access to specific product functions and features. System security does not override or replace the member or master passwords.

The product security features provide flexible security options, while coexisting with any other security systems already used in your data processing operation.

You can secure BMS/TS functions based on one or more of the following factors:

- Transaction Code,
- Command,
- Member Type,
- Set Name.

Security is controlled by operator ID (*OPID) records. It is not necessary to create an *OPID record for each user. You can create a default record for users who share similar access, and create individual *OPID records only for users with special access needs.

You can enable security online (via Product Options) or via the GTBUTIL batch utility.

Understanding *OPID Records

All operator ID records have the Set/Type of *OPID. If security is enabled, when a user signs on, BMS/TS checks to see if there's an *OPID record for that user. If there is not, the user is given the access specified in the default record.

The *OPID member name and sign–on method are determined as follows:

- If you are using BMS/TS security alone, the *OPID member name is the name with which the user will sign on to BMS/TS. When accessing the BMS/TS System Menu, the user must select the Signon option, and enter the member name. BMS/TS then refers to that *OPID record to determine access, rather than the default record. If the user does not sign–on to BMS/TS, the system uses the default record to determine access.

- If you are using CICS security, the *OPID member name must match the CICS User ID. There is no need to access the BMS/TS Signon screen, because the user has already signed on to CICS.

You can provide additional security by specifying a password for the *OPID record. The user is forced to key the password when signing on to BMS/TS.

The *OPID record entries specify whether a user request is “accepted” or “rejected”. The order of the entries is important! The first matching entry will determine whether the requested action is accepted or rejected. We recommend that the last entry in each *OPID record should be an “A” in the A/R field. This entry tells the system to accept any request that has not been secured (rejected).

Requests must exactly match the entries on the Operator Security Editor screen. Qualifiers that are left blank on the editor screen match any value when the record is scanned.

Profiles

The Profile function gives users separate modules for customized system messages and system maps, and enables them to access their own help screens from a separate file.

Profile is especially useful for multilingual environments where there is a need to produce help displays, maps and messages in another language.

Setting Up System Security

1. Create the Product Administrator’s *OPID Record

   The first *OPID record you create is for the Product Administrator. This record contains only an “A” in the A/R field. This gives you (the Product Administrator) access to all BMS/TS functions and commands.

2. Create the Default *OPID Record

   Next, create the default *OPID record. This determines the security access for all users who do not have an individual *OPID record. Reject any transactions, commands, set types and set names that you do not want users to access. The last line of this record is an “A” in the A/R field, accepting any request that has not been specifically rejected.

   If you are using the BMS/TS Signon feature, this default record should only allow access to the Signon command. This forces the user to sign on to BMS/TS. After signing on, the user’s *OPID record determines access. You will need to create an *OPID record for each user.

3. Create Individual User *OPID Records

   Create the individual *OPID records for users with special access needs. Again, list all “rejected” entries first. The last line of the *OPID record must be an “A” in the A/R field, to accept all other requests that have not been specifically rejected.

If two or more records are similar, use the Utility screen to eliminate repetitive keying. Copy the original record to the new name(s), then edit the new record(s) as needed.

4. Activate BMS/TS Security

After you have created the necessary *OPID records, you are ready to activate internal security. You can activate security online or via batch.

- Online: The security switch is located on the Product Options screen. Type a "Y" in the Force Security field. Press Enter to save the change, then F3 to exit the Product Options. Security is enabled immediately.
- Batch: Execute the batch utility GTBUTIL (using ALTER, SIGNON=YES). Security is enabled as soon as the batch job is finished.

Security Options

This section explains how to secure BMS/TS menus and certain administrative functions.

Security Options for BMS/TS Menus

Some BMS/TS menus may be required for only a few users. To restrict access by anyone else, you could restrict each menu function separately. But you also have the option of restricting the entire menu. When you do that, the menu option does not appear on the user's screen. Rather than getting an "access denied" message, the user never sees the option.

For example, you may decide to restrict most users from access to any of the Administration functions. If you restrict each function separately, the user will still see the Administration option displayed on the System Menu. If you restrict the entire menu, the Administration option never appears on the users' System Menu.

When an entire menu is restricted, the option is deleted from the previous screen. The remaining menu options are renumbered, so the user doesn't see that a menu item is missing.

If you want to restrict access to an entire menu, contact GT Software Technical Support for assistance.

Security Options for the Retrieve (Transfer) Function

The File Transfer/Member Retrieve function moves members between libraries, and even between GT Software products. To prevent unauthorized access to special libraries, there are special security options.

The Retrieve function uses the sub−commands Copy, Fetch, and Move. BMS/TS uses special commands to accept or reject these functions. These special commands include:

- **RETC** Retrieve−Copy
- **RETF** Retrieve−Fetch
- **RETM** Retrieve−Move

You can use these special commands in the Command field. For example, you could reject access to the Fetch sub−command but allow the Copy and Move commands.

Retrieve DDName Validation

Keep in mind that security validates each DDName used during the Retrieve function; i.e. the “To” and “From” library prefixes.

To prevent access to a particular library prefix, leave the Command field blank. Enter Type=*File, Set=xx, where “xx” is the prefix of the protected DDName. This restricts sending from or receiving into the specified library.

You can also use special commands to restrict access to a specific library, without restricting each function (Copy, Fetch and Move) individually. The commands are:

- **RETTO** Retrieve to
- **RETFM** Retrieve from

If you want to protect only against moving a member into a particular library, type the command retto in the user’s *OPID record. If you want to control the “from” library, use the command retfm.

Security Options for the Merge Function

The Merge function affects the Transaction Monitor Table, and any *SYSTEM members. To prevent unauthorized access, internal security uses special commands to accept or reject these functions. These special commands include:

Command	Type	Set Name	Description
MERGEF	*SYSTEM	*xxxxxxxx*	Merge from Monitor Table
MERGET	*SYSTEM	*xxxxxxxx*	Merge from Monitor Table
TRANF	*TRANS	*xxxx*	Merge from transaction
TRANT	*TRANS	*xxxx*	Merge to transaction
MFM	*FILE	*xx*	Merge from $FILE
MTO	*FILE	*xx*	Merge to $FILE

Using these commands, you have several options for preventing unauthorized access to Transaction Monitor Tables.

- The MERGEF and MERGET commands restrict merges to and from specific *SYSTEM tables. You identify the specific table in the Set Name field.
- The TRANF and TRANT commands restrict merges to and from specific transactions. In the Set Name field, you identify the transaction you want to restrict.
- The MFM and MTO commands restrict merges by file prefix. You specify the 2–character file prefix in the Set Name field.

Security Options for Profile Security

The Profile function provides the user with separate modules for customized system messages and system maps, and enables users to access their own help screens from a separate file. Profile is especially useful for multilingual environments where there is a need to produce help displays, maps and messages in another language.

By specifying Profile entries in the *OPID Record, you alert the security system that this user will utilize different internal modules. The three Profile options are specified by entering the following *OPID Record entries:

Field	Description
A/R	Type P to indicate a Profile entry.
Tran	Type *USE.
Command	Must contain one or more of the following module names: • **MSGLANG** –– Indicates that this user has a different internal message module. The new module is identified as GTB$MSGx, where “x” is the suffix specified in the Set Name field. • **MAPLANG** –– Indicates that this user has a different mapset for customized system maps. The new mapset is identified as GTBMAPx, where “x” is the suffix specified in the Set Name field. • **INTLIB** –– Indicates that help screens for this user reside in a different file. The file prefix is identified in the Set Name field. In order to access the alternate help screens, all monitored transactions must specify FILE=** in the Transaction Monitor Table. When there is a help request for a transaction with FILE=**, the system checks the active *OPID record to see if it points to an alternate INTLIB. If it does, the help is displayed from the alternate help library. If there is no INTLIB specified on the *OPID record, the system uses the INTLIB specified in the *ADMIN record.
Type	Used only with the INTLIB command. If you specified the INTLIB command, you must type *FILE in this field.
Set Name	Dependent on the Command field entry. • When used with MSGLANG or MAPLANG commands, this entry contains the single–character suffix to be appended to the modules. • Used with the INTLIB command, this entry contains the two–character file prefix for the file that contains the help displays for this user.

Samples of Common Security Record Entries

This section contains some examples of commonly−used security entries.

Reject Access to Update Specific Member Types

A/R	Tran	Command	Type	Set Name	Comments
R		UPDATE	*COMPANY	POLICY	Rej. changes to policies
R		UPDATE	*SYSTEM		Rej. Changes to monitor

This user cannot update the member POLICY in the *COMPANY Set/Type; user also cannot update any member with the Set/Type *SYSTEM. This user can update members with other Set/Types, such as *MENU or *JCL.

Reject Access to Update Specific Members

A/R	Tran	Command	Type	Set Name	Comments
R		UPDATE	*JCL	PAYCODE	Restr. Pay Code JCL
R		UPDATE	*MENU	PAYSCHED	Restr. Pay Code Menu

This user cannot update the “Paycode” JCL or the “Paysched” menu. This user can update other members, and other Set/Types.

Reject All Access to the Transaction Monitor Table

A/R	Tran	Command	Type	Set Name	Comments
R			*SYSTEM		

This user cannot access the Transaction Monitor Table for any reason.

Restrict Access to the Transaction Monitor Table

A/R	Tran	Command	Type	Set Name	Comments
R		DELETE	*SYSTEM		
R		ADD	*SYSTEM		

This user cannot delete an existing Transaction Monitor Table, or add a new one. However, this user can access existing Transaction Monitor Tables for other purposes, such as updating.

Restrict Access to the *ADMIN Record

A/R	Tran	Command	Type	Set Name	Comments
R			*ADMIN		

This user cannot access the *ADMIN record. The Product Settings menu option will appear on the Administration Menu, but if this user tries to access those screens, the system will display an error message.

Restrict Access to the Administration Menu

A/R	Tran	Command	Type	Set Name	Comments
R		MENMENU	*MENU	GTAM06	

This user cannot access any option on the Administration Menu. Further, the Administration option will not display on the System Menu.

Default Record Allowing Signon Access Only

A/R	Tran	Command	Type	Set Name	Comments
A		SIGNON			
R					

When used as the default *OPID record, this forces all users to sign on to BMS/TS before continuing. Each user must have an individual *OPID record.

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Change Product Options

You can update the *ADMIN record in order to change various BMS/TS options, such as:

- Managing multiple libraries,
- Controlling security switches, dates, and editing patterns,
- Supporting expanded function keys,
- Using a master password,
- Using other GT products and the Transaction Monitor.

Note: Because the installation of BMS/TS creates the *ADMIN record, you may have determined several of the above options when you installed BMS/TS.

Enter Master Password

If you have created a master password, you must enter it before you can access the Change Product Options screen.

Note: This screen only displays if you have already created a master password. If you have not yet defined a master password, BMS/TS proceeds directly to the Change Product Options screen.

1. From the Main Menu, select the Product option and press Enter. The Product Menu displays (view sample).
2. From the Product Menu, select the Change Product Options option and press Enter. The Change Product Options Security screen displays (view sample). The cursor moves to the Master Password field.
3. Type the master password.
4. Press Enter. BMS/TS displays the Change Product Options screen (view sample).

Change Product Options

Changing product options lets you modify your *ADMIN record by editing or defining the master password, library prefixes, and security and date options.

Perform the following steps to change product options.

When you complete the Change Product Options Security screen, the Change Product Options screen appears (view sample).

1. Press Tab to move to the first field you want to modify.

2. Type over the existing information.

3. Repeat steps 1 and 2 to modify all desired fields.

4. At this point you have two options:
   - If you want to edit or define hot keys for BMS/TS help screens, and the Transaction Monitor prefix, press F2 to access the Special Product Options screen.
   - To save the updated settings, press Enter, then press F3 to exit this screen. You **must** press Enter or your changes will not be saved.

Note: All changes to the *ADMIN record are activated immediately at the terminal where you make your changes. However, to ensure that the current changes take effect for all BMS/TS users, have your programmers exit both BMS/TS and the current CICS session, then re–enter CICS and BMS/TS.

Change Special Product Options

Changing special product options lets you modify your *ADMIN record by editing or defining hot keys for BMS/TS help screens, and the Transaction Monitor prefix.

If you press F2 from the Change Product Options screen, the Special Product Options screen appears (view sample).

1. Press Tab to move to the first field you want to modify.

2. Type over the existing information.

3. Repeat steps 1 and 2 to modify all desired fields.

4. At this point you have two options:
   - If you want to review or update the Master Password, library prefixes, and/or security and date options, press F2 to return to the Change Product Options screen.
   - To save the updated settings, press Enter, then press F3 to exit this screen. You **must** press Enter or your changes will not be saved.

Note: All changes to the *ADMIN record are activated immediately at the terminal where you make your changes. However, to ensure that the current changes take effect for all BMS/TS users, have your programmers exit both BMS/TS and the current CICS session, then reenter CICS and BMS/TS.

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Customizing BMS/TS

Administrator's Guide Home
BMS/TS Doc. Home

Customizing with User Exits

You can write user exits for use with BMS/TS.

User exits refer to points within BMS/TS at which BMS/TS processing is suspended and control is passed through links to user defined programs. BMS/TS lets you to write user exits so that you can branch from BMS/TS processing, perform other routines, then return to BMS/TS.

For example, users typically write routines to perform internal functions such as system accounting, setting up and deleting special temporary storage records, counting help calls for fields or screens, and security checks. Users also write routines to perform external functions such as making inquiries of databases, reading files outside BMS/TS, and calling security systems.

Supported User Exit Points

The following table lists the user exit points supported by BMS/TS.

User Exit Program Name	Description
GTB@HLPt *	Help Initiation
GTB@HCPt *	Help Termination
GTB@SCTt *	Internal Security
GTB@SUBX	Job Submission
GTB@FLDX	Field Name Validation

* The “t” represents the code for the type of program. For command–level programs, user exits must end in the letter "X": for example, GTB@HLPX. Macro–level programs must end in the letter "M": for example, GTB@HLPM.

User Exit Processing

User exits function as outlined below:

1. At a particular point in processing, BMS/TS checks to see it should make a call to a user defined routine. BMS/TS uses link edit WXTRN to determine if an exit point has been activated. The user exit stub (for command–level) or the macro–level program resolves the WXTRN.

2. For command−level programs only: if a call is to be made, BMS/TS uses the COMMAREA (common passed area) to pass control and specified parameters to the user routine.

3. The user defined program executes its functions.

4. The user program returns control to BMS/TS along with a return code. The user program may also pass parameters to BMS/TS.

User Exit Parameters

This section describes the parameters in the COMMAREA that BMS/TS can pass to the user−defined routine, and/or that the user−defined routine can pass to BMS/TS.

For user exits written in command−level code, BMS/TS uses the COMMAREA to pass control and specified parameters to the user routine.

> **Note:** If you are using CICS version 3.2 and above, you must code the exit in command−level code. If you are using CICS version 3.1 and below, you can code the exits in either command−level or macro−level code.

The table below describes the fields in the COMMAREA.

Field Name	Field Type/Size		Description
	COBOL	Assembler	
GT$−UXCLN	PIC S9(4) COMP.	H	The length of the user exit area (COMMAREA). BMS/TS passes this field to the user routine.
GT$−URETC	PIC X.	C	The return code that is passed from BMS/TS to the user routine, and/or from the user routine to BMS/TS. Valid values are: **X'00'** Normal return **C'R'** Reject the request **C'A'** Abort the transaction
GT$−UACTC	PIC X.	C	The user action code. This field is passed from BMS/TS to the user routine, and/or from the user routine to BMS/TS. Valid values are: **X'00'** Default value (empty) **C'S'** Save the current screen contents when BMS/TS enters the exit **C'R'** Restore the screen using the contents that were saved

GT$–UXPGM	PIC X(8).	CL8	The user exit program name. BMS/TS passes this field to the user routine.
GT$–UAPGM	PIC X(8).	CL8	The name of the BMS/TS calling program. BMS/TS passes this field to the user routine.
GT$–UATRN	PIC X(4).	CL4	The transaction code ID of BMS/TS. BMS/TS passes this field to the user routine.
GT$–UXTRN	PIC X(4).	CL4	The requesting user's transaction code ID. This is the transaction code of the user's system before the user pressed the help key. BMS/TS passes this field to the user routine.
GT$–USET	PIC X(8).	CL8	Data that is used as the *FIELD member. This is the current mapset name of the help map BMS/TS expects to display. BMS/TS passes this field to the user routine.
GT$–UMAP	PIC X(8).	CL8	Data that is used to search the *FIELD for a matching map name. This is the current map name of the help map BMS/TS expects to display. BMS/TS passes this field to the user routine.
GT$–UHTYP	PIC X(8).	CL8	The help map set/type. This is the map name of the alternative help map as specified by the user routine. The user routine passes this field to BMS/TS.
GT$–UHMBR	PIC X(8).	CL8	The member name of an alternative help map, as specified by user routine. The user routine passes this field to BMS/TS.
GT$–OPID	PIC X(8).	CL8	The CICS operator ID. BMS/TS passes this field to the user routine.
GT$–UCLOC	PIC S(4) COMP–4.	CL2	The location of the cursor, relative to character 0, on the application screen when the help key was pressed. BMS/TS passes this field to the user routine.
GT$–UAID	PIC X.	C	The identity of the PF key that requested help. BMS/TS passes this field to the user routine.

COMMAREA Examples

The following figure shows the COMMAREA used by an Assembler exit.

```
*            COMMON PASSED AREA
```

```
GT$UEXIT    DS     OF     COMMON AREA PASSED BY GTS
GT$UXCLN    DS     H      LENGTH OF USER EXIT A
GT$URETC    DS     C      RETURN CODE
*                           X'00'NORMAL RETURN
*                           C'R' REJECT THE REQUEST
*                           C'A' ABORT THE TRANSACTION
GT$UACTC    DS     C      USER ACTION CODE
*                           C'S' SAVE CALL
*                           C'R' RESTORE
GT$UXPGM    DS     CL8    USER EXIT PROGRAM NAME
GT$UAPGM    DS     CL8    BMS/TS CALLING PROGRAM
GT$UATRN    DS     CL4    BMS/TS TRAN CODE
GT$UXTRN    DS     CL4    USER TRAN CODE REQUEST
GT$USET     DS     CL8    USED AS *FIELD MBR
GT$UMAP     DS     CL8    USED AS *FIELD INTERNAL
GT$UHTYP    DS     CL8    HELP SCREEN SET/TYPE
GT$UHMBR    DS     CL8    HELP SCREEN MEMBER
GT$OPID     DS     CL8    CICS OPERATOR ID
GT$UCLOC    DS     CL2    CURSOR LOCATION
GT$UAID     DS     C      AID THAT REQUESTED HELP
```

The following figure shows the COMMAREA used by a COBOL exit.

```
LINKAGE SECTION
*     COMMON PASSED AREA
01    DFHCOMMAREA.
      05   GT$-UXCLN     PIC S9(4) COMP.
      05   GT$-URETC     PIC X.
      05   GT$-UACTC     PIC X.
      05   GT$-UXPGM     PIC X(8).
      05   GT$-UATRN     PIC X(4).
      05   GT$-UAPGM     PIC X(8).
      05   GT$-UXTRN     PIC X(4).
      05   GT$-USET      PIC X(8).
      05   GT$-UMAP      PIC X(8).
      05   GT$-UHTYP     PIC X(8).
      05   GT$-UHMBR     PIC X(8).
      05   GT$-OPID      PIC X(8).
      05   GT$-UCLOC     PIC S(4) COMP-4.
      05   GT$-UAID      PIC X.
```

Link–Edit JCL

This section provides examples of the link–edit JCL you must write to activate a command or macro–level user exits. These JCL samples use "dummy" program names to represent the actual information you will include in the JCL. The following table describes the program names that appear in these JCL **samples**:

Program Name	Description
GTB*ccccc*	Indicates the BMS/TS calling program, created by linking the include modules. Replace the '*ccccc*' with the actual program identifier.
GTB$*iiii*	Indicates the BMS/TS modules to be included. Replace 'GTB$*iiii*' with one or more include module program names. See the specific user exit discussions for

examples.

GTB@*xxx*	Identifies the BMS/TS exit point. Replace '*xxx*' with the appropriate exit identifier.
GTB@*xxx*X	Name of the user's CICS command−level program.
GTB@*xxx*C	Name of the user exit stub. This is provided with the BMS/TS object library.
GTB@*xxx*M	Indicates the user's CICS macro−level program.

Command−Level Examples

The following figures show examples of JCL statements for command−level programs. Keep in mind that these are only examples: your JCL must be specific to the files and libraries used at your company.

OS/390 Command−Level JCL Example

```
//LNKOSC    JOB
//LNKED     EXEC  PGM=IEWL,PARM'LIST,XREF,NOLET'
//SYSPRINT  DD    SYSOUT=A
//SYSUT1    DD    UNIT=SYSDA,SPACE=(1024,(100,10))
//SYSLMOD   DD    DSN=GTS.LOADLIB,DISP=SHR
//SYSLIB    DD    DSN=GTB.OBJLIB,DISP=SHR
//SYSLIN    DD    DSN=GTB.USER.OBJLIB,DISP=SHR
//SYSIN     DD    *
 INCLUDE    SYSLIB(GTBcccc)    * BMS/TS CALLING PROGRAM
 INCLUDE    SYSLIB(GTB$iiii)   * Enter BMS/TS include
 .                             * statements here
 INCLUDE    SYSLIB(GTB@xxxc)   * BMS/TS USER EXIT STUB
 NAME              GTBcccc(R)
/*
//LNKED2    EXEC  PGM=IEWL,PARM'LIST,XREF,NOLET'
//SYSPRINT  DD    SYSOUT=A
//SYSUT1    DD    UNIT=SYSDA,SPACE=(1024,(100,10))
//SYSLMOD   DD    DSN=GTS.LOADLIB,DISP=SHR
//SYSLIN    DD    DSN=USER.OBJLIB,DISP=SHR
//SYSIN     DD    *
 NAME             GTB@xxxx(R) * USER PROGRAM
/*
/&
```

VSE Command−Level JCL Example

```
// JOB LNKVSEC
// EXEC PROC=GTBLIBS
// OPTION CATAL
   INCLUDE GTBccccc     * BMS/TS CALLING PROGRAM
   INCLUDE GTB$iiii     * BMS/TS INCLUDE MODULES
   INCLUDE                    * Enter BMS/TS include
     .                        * statements here
   INCLUDE GTB@xxxc     * BMS/TS USER EXIT STUB
// EXEC LNKEDT
*
// OPTION CATAL
   INCLUDE GTB@xxxx     * USER PROGRAM
// EXEC LNKEDT
```

```
/*
/&
```

Macro–level Examples

The following figures show examples of JCL statements for macro–level programs. Keep in mind that these are only examples; your JCL must be specific to the files and libraries used at your company.

OS/390 Macro–level JCL Example

```
//LNKOSM    JOB
//LNKED     EXEC   PGM=IEWL,PARM'LIST,XREF,NOLET'
//SYSPRINT  DD     SYSOUT=A
//SYSLMOD   DD     DSN=GTS.LOADLIB,DISP=SHR
//SYSLIB    DD     DSN=GTB.OBJLIB,DISP=SHR
//CICSLIB   DD     DSN=CICS.LOADLIB,DISP=SHR
//SYSUT1    DD     UNIT=SYSDA.SPACE=(1024,(100,10))
//SYSLIN    DD     DSN=GTB.OBJLIB,DISP=SHR
//                 DSN=USER.OBJLIB,DISP=SHR
//SYSIN     DD     *
 INCLUDE    SYSLIB(GTB@xxxm)       * USER Macro-level PROGRAM
 NAME               GTBcccc(R)     * BMS/TS CALLING PROGRAM
/*
/&
```

VSE Macro–level JCL Example

```
// JOB LNKVSEM
// EXEC PROC=GTBLIBS
// OPTION CATAL
   INCLUDE GTBccccc      * BMS/TS CALLING PROGRAM
   INCLUDE GTB$iiii      * BMS/TS INCLUDE MODULES
                         * Enter BMS/TS include
 .                       * statements here
   INCLUDE GTB@xxxm      * USER Macro-level PROGRAM
// EXEC LNKEDT
/*
/&
```

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Customizing with User Exits

Administrator's Guide Home
BMS/TS Doc. Home

GTB@FLDX, User Exit for Field Name Validation

During this exit, you can perform any CICS command–level function except for screen I/Os or functions which would release any resources.

> **Note:** This exit only supports command–level code.

The field name validation user exit lets you to examine field names entered during the Field Definition step of map maintenance to verify that field names coincide with your company's naming conventions. This lets you supplement the BMS/TS field name validation, or reject a field name, if desired. BMS/TS calls this user exit for each named field in a screen, whether the name is new or unchanged.

BMS/TS will pass the exit to all named fields for all language types. However, you must ensure that the name validation allows for all language syntax.

> **Note:** You cannot modify the name entered. Your option is only to accept or reject the name as entered.

Passing Parameters

For field name validation user exits written in command–level code, BMS/TS uses the COMMAREA to pass control and specified parameters to the user routine. The COMMAREA also contains return code fields to notify BMS/TS, upon return from the exit, what action to take.

See User Exit Parameters for a description of all fields that can appear in the COMMAREA.

Writing GTB@FLDX, Command–Level User Exit for Field Name Validation

Perform the following steps to write the command–level user exit for field name validation.

1. Code the GTB@FLDX exit routine as a CICS program. Remember, the standard COMMAREA transfers information to and from the user exit.

   Note: The name of the user exit stub ends in a "C": GTB@FLDC.

2. Assemble the program, making certain to include the COMMAREA.

3. Create links to pass control from BMS/TS to your own program. To do this, link–edit GTB@FLDX as a standalone load module in CICS. Be sure to create a CSD entry for this program.

4. Re–link the BMS/TS calling program. Also, remember to include a statement for GTB@FLDC.

GTB@FLDX Example

A sample Assembler program appears below. Keep in mind that this is only an example: your program must be specific to the files and libraries used at your company.

```
*************************************************************
* SAMPLE FIELD EXIT COMMAND LEVEL CONVENTIONS
*
*************************************************************

 TITLE 'GT@FLD - USER INTERFACE FOR BMS/TS '

WORKAREA DSECT
         DS H LENGTH OF THE IS AREA
RTRNCODE DS X RETURN CODE
* PASSED BACK
* X'00' NORMAL RETURN
* > 0 REJECT AND CANCEL
FLDNAMEL DS F FIELD NAME LENGTH
FLDNAME  DS CL30 FIELD NAME TO BE VALIDATED
*
WORK2 EQU 2
GTB@FLDX CSECT ,
         L WORK2,DFHEICAP PICKUP ADDR OF PARMS
         USING WORKAREA,WORK2 SETUP BASE FOR COMMAREA
*** PUT YOUR EDITS HERE
***
*** USE MVI RTRNCODE,X'00' FOR GOOD RETURN CODE
*** USE MVI RTRNCODE,X'FF' FOR BAD RETURN CODE
EXITPGM DS 0H
        EXEC CICS RETURN
*
        END
```

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

GTB@HCPt, User Exit for Help Termination

BMS/TS calls the user exit for help termination after displaying the requested help screen.

> **Note:** For command–level programs, this user exit must end in the letter "X": GTB@HCPX. Macro–level programs must end in the letter "M": GTB@HCPM.
>
> Also, remember that if you are using CICS version 3.2 and above, you must code the exit in command–level code. If you are using CICS version 3.1 and below, you can code the exit in either command–level or macro–level code.

The user exit for help termination let you write a routine to perform a variety of accounting procedures, including counting the number of times a field required help, the total number of calls to the help processor, as well as freeing or retrieving previously acquired or saved storage.

This exit also provides system access to temporary storage and internal accounting after the completion of a help function. You can use this exit in conjunction with the GTB@HLPC exit to use any temporary storage created before the HELP function was invoked.

Passing Parameters

For help termination user exits written in command–level code, BMS/TS uses the COMMAREA to pass control and specified parameters about the help request to the user routine. The COMMAREA also contains return code fields to notify BMS/TS, upon return from the exit, what action to take.

See User Exit Parameters for a description of all fields that can appear in the COMMAREA.

> **Note:** Changes made to the common parameter area have no effect on the return process.

Writing GTB@HCPX, Command–Level User Exit for Help Termination

Perform the following steps to write the command–level user exit for help termination.

1. Code the GTB@HCPX exit routine as a CICS program. Remember, the standard COMMAREA transfers information to and from the user exit.

   Note: The name of the user exit stub ends in a "C": GTB@HCPC.

2. Assemble the program, making certain to include the COMMAREA.

3. Create links to pass control from BMS/TS to your own program. To do this, link–edit GTB@HCPX as a standalone load module in CICS. Be sure to create a CSD entry for this program.

4. Re–link the BMS/TS calling program –– in this case, GTBHCUP, the BMS/TS clean–up program. GTBHCUP de–allocates storage and deletes temporary records before returning control to BMS/TS. Also, remember to include a statement for GTB@HCPC.

GTB@HCPC Examples

Below are sample control statements for linking the BMS/TS load modules.

OS/390 Example

```
INCLUDE SYSLIB(GTBHCUP)
INCLUDE SYSLIB(GTB$PCTM)
INCLUDE SYSLIB(GTB$$EI1)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$PCL)
INCLUDE SYSLIB(GTB$$PFL)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB@HCPC) * User exit stub
NAME    GTBHCUP(R)
```

VSE Example

```
INCLUDE GTBHCUP
INCLUDE GTB$PCTM
INCLUDE GTB$$EI1
INCLUDE GTB$$ERG
INCLUDE GTB$$PCL
INCLUDE GTB$$PFL
INCLUDE GTB$$SRG
INCLUDE GTB@HCPC * User exit stub
```

Writing GTB@HCPM, Macro–Level User Exit for Help Termination

Perform the following steps to write the macro–level user exit for help termination.

1. Code the GTB@HCPM exit routine as a CICS program.

   Note: When using macro–level exits with the Assembler language, CICS accesses data through register 1, which points to the standard parameter list. The first address in this list points to the

pass area.

2. Assemble the program.

3. Create links to pass control from BMS/TS to your own program. To do this, link your program as part of the total GT load module with which the program is associated.

4. Re–link the BMS/TS calling program –– in this case, GTBHCUP, the BMS/TS cleanup program. GTBHCUP de–allocates storage and deletes temporary records before returning control to BMS/TS. Also, remember to include a statement for GTB@HCPM. You should assemble and place your user exit in your BMS/TS (or other) OBJLIB.

GTB@HCPM Examples

Below are sample control statements for linking the BMS/TS load modules.

OS/390 Example

```
INCLUDE SYSLIB(GTBHCUP)
INCLUDE SYSLIB(GTB$PCTM)
INCLUDE SYSLIB(GTB$$EI1)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$PCL)
INCLUDE SYSLIB(GTB$$PFL)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB@HCPM) * Macro-level user exit
NAME    GTBHCUP(R)
```

VSE Example

```
INCLUDE GTBHCUP
INCLUDE GTB$PCTM
INCLUDE GTB$$EI1
INCLUDE GTB$$ERG
INCLUDE GTB$$PCL
INCLUDE GTB$$PFL
INCLUDE GTB$$SRG
INCLUDE GTB@HCPM *Macro-level user exit
```

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

GTB@HLPt, User Exit for Help Initiation

BMS/TS calls the user exit for help initiation prior to displaying the requested help display.

> **Note:** For command–level programs, this exit must end in the letter "X": GTB@HLPX. Macro–level programs must end in the letter "M": GTB@HLPM.
>
> Also, remember that if you are using CICS version 3.2 and above, you must code the exit in command–level code. If you are using CICS version 3.1 and below, you can code the exit in either command–level or macro–level code.

The user exit for help initiation has three capabilities:

- It enables you to display different help screens based on an operator ID. This lets you vary the help information that is provided to individual IDs or to groups of IDs. For example, you might vary the help screens that display for the same function based on user type: beginner, novice, expert, etc.
- It can replace the GT$USER/GT$UMAP entries (help set and map names) to present an alternate display.
- It provides access to temporary storage before BMS/TS initiates a help function. In conjunction with the user exit for Help Termination, you can use this exit to initialize temporary storage for use by BMS/TS after the HELP function is invoked.

Passing Parameters

For help initiation user exits written in command–level code, BMS/TS uses the COMMAREA to pass control and specified parameters about the help request to the user routine. The COMMAREA also contains return code fields to notify BMS/TS, upon return from the exit, what action to take.

See User Exit Parameters for a description of all fields that can appear in the COMMAREA.

GT$URETC Return Field

BMS/TS provides a return field (GT$URETC) that lets the exit deny or allow a help request. GT$URETC is passed from BMS/TS to the user routine, and/or from the user routine to BMS/TS.

GT$URETC Value	Description
x'00'	Normal return. Allows the help request. The help screen displays.
c'R'	Reject help request. Denies the help request. No help screen displays.
c'A'	Abort help request. Cancels the help request. No help screen displays.

Note: After the return code, CICS calls the clean up program GTBHCUP to de−allocate storage and delete temporary records, before returning control to BMS/TS. For more information refer to GTB@HCPt, User Exit for Help Termination.

Displaying an Alternate Help Screen

If desired, you can also instruct this user exit to replace the GT$USER/GT$UMAP entries (help set and map names) in the COMMAREA. This enables you to display an alternate help screen. To do this, add the following entries to your JCL:

GT$UHTYP The name of the mapset that BMS/TS should display.

GTUHMBR The name of the map that BMS/TS should display.

For example, the settings GT$UHTYP = BILLING and GT$UHMBR = MAP1, instruct BMS/TS to display the MAP1 map from the BILLING mapset.

Writing GTB@HLPX, Command−Level User Exit for Help Initiation

Perform the following steps to write the command−level user exit for help initiation:

1. Code the GTB@HLPX exit routine as a CICS program. Remember, the standard COMMAREA transfers information to and from the user exit.

   Note: The name of the user exit stub ends in a "C": GTB@HLPC.

2. Assemble the program, making certain to include the COMMAREA.

3. Create links to pass control from BMS/TS to your own program. To do this, link−edit GTB@HLPX as a standalone load module in CICS. Be sure to create a CSD entry for this program.

4. Re−link the BMS/TS calling program −− in this case, GTBUHLP, the BMS/TS main help processor. Remember to include a statement for GTB@HLPC.

GTB@HLPX Examples

Below are sample control statements for linking the BMS/TS load modules.

OS/390 Example

```
INCLUDE SYSLIB(GTBUHLP)
INCLUDE CICSLIB(DFHEAI)
INCLUDE SYSLIB(GTB@HLPC) * User exit stub
INCLUDE CICSLIB(DFHEAIO)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB$PCTM)
INCLUDE SYSLIB(GTB$BPCK)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$FCL)
INCLUDE SYSLIB(GTB$$IM1)
INCLUDE SYSLIB(GTB$$MOR)
INCLUDE SYSLIB(GTB$$PCL)
INCLUDE SYSLIB(GTB$$PFL)
INCLUDE SYSLIB(GTB$$PPL)
INCLUDE SYSLIB(GTB$$RHN)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$HDLD)
INCLUDE SYSLIB(GTB$$IM2)
INCLUDE SYSLIB(GTB$$RED)
INCLUDE SYSLIB(GTB$$SBF)
INCLUDE SYSLIB(GTB$$COS) * For CICS 3.1 and above, use GTB$$CES
INCLUDE SYSLIB(GTB$$EXP)
INCLUDE SYSLIB(GTB$$OOS) * For CICS 3.1 and above, use GTB$$OES
INCLUDE SYSLIB(GTB$$TOS) * For CICS 3.1 and above, use GTB$$TES
NAME    GTBUHLP(R)
```

VSE Example

```
INCLUDE GTBUHLP
INCLUDE DFHEAI
INCLUDE DFHEAIO
INCLUDE GTB$SCTY
INCLUDE GTB$PCTM
INCLUDE GTB$BPCK
INCLUDE GTB$$ERG
INCLUDE GTB$$FCL
INCLUDE GTB$$IM1
INCLUDE GTB$$MOR
INCLUDE GTB$$PCL
INCLUDE GTB$$PFL
INCLUDE GTB$$PPL
INCLUDE GTB$$RHN
INCLUDE GTB$$SRG
INCLUDE GTB$HDLD
INCLUDE GTB$$IM2
INCLUDE GTB$$RED
INCLUDE GTB$$SBF
INCLUDE GTB$$CDO
INCLUDE GTB$$EXP
INCLUDE GTB$$ODO
INCLUDE GTB$$TDO
INCLUDE GTB@HLPC * User exit stub
```

Writing GTB@HLPM, Macro−Level User Exit for Help Initiation

Perform the following steps to write the macro−level user exit for help initiation.

1. Code the GTB@HLPM exit routine as a CICS program.

   Note: When using macro−level exits with the Assembler language, CICS accesses data through register 1, which points to the standard parameter list. The first address in this list points to the pass area.

2. Assemble the program.

3. Create links to pass control from BMS/TS to your own program. To do this, link your program as part of the total GT load module with which the program is associated.

4. Re−link the BMS/TS calling program −− in this case, GTBUHLP, the BMS/TS main help processor. Remember to include a statement for GTB@HLPM. You should assemble and place your user exit in your BMS/TS (or other) OBJLIB.

GTB@HLPM Examples

Below are sample control statements for linking the BMS/TS load modules.

OS/390 Example

```
INCLUDE SYSLIB(GTBUHLP)
INCLUDE CICSLIB(DFHEAI)
INCLUDE SYSLIB(GTB@HLPM) * Macro-level user exit
INCLUDE CICSLIB(DFHEAIO)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB$PCTM)
INCLUDE SYSLIB(GTB$BPCK)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$FCL)
INCLUDE SYSLIB(GTB$$IM1)
INCLUDE SYSLIB(GTB$$MOR)
INCLUDE SYSLIB(GTB$$PCL)
INCLUDE SYSLIB(GTB$$PFL)
INCLUDE SYSLIB(GTB$$PPL)
INCLUDE SYSLIB(GTB$$RHN)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$HDLD)
INCLUDE SYSLIB(GTB$$IM2)
INCLUDE SYSLIB(GTB$$RED)
INCLUDE SYSLIB(GTB$$SBF)
INCLUDE SYSLIB(GTB$$COs) * For CICS 3.1 and up, use GTB$$CES
INCLUDE SYSLIB(GTB$$EXP)
INCLUDE SYSLIB(GTB$$OOS) * For CICS 3.1 and above, use GTB$$OES
INCLUDE SYSLIB(GTB$$TOS) * For CICS 3.1 and above, use GTB$$TES
NAME GTBUHLP(R)
```

VSE Example

```
INCLUDE (GTBUHLP)
INCLUDE GTBUHLP
```

```
INCLUDE DFHEAI
INCLUDE DFHEAIO
INCLUDE GTB$SCTY
INCLUDE GTB$PCTM
INCLUDE GTB$BPCK
INCLUDE GTB$$ERG
INCLUDE GTB$$FCL
INCLUDE GTB$$IM1
INCLUDE GTB$$MOR
INCLUDE GTB$$PCL
INCLUDE GTB$$PFL
INCLUDE GTB$$PPL
INCLUDE GTB$$RHN
INCLUDE GTB$$SRG
INCLUDE GTB$HDLD
INCLUDE GTB$$IM2
INCLUDE GTB$$RED
INCLUDE GTB$$SBF
INCLUDE GTB$$CDO
INCLUDE GTB$$EXP
INCLUDE GTB$$ODO
INCLUDE GTB$$TDO
INCLUDE GTB@HLPM * Macro-level user exit
```

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Customizing with User Exits

Administrator's Guide Home
BMS/TS Doc. Home

GTB@SCTt, User Exit for Internal Security

The user exit for internal security checks to see if the operator is allowed to use the current command or transaction. BMS/TS calls this exit prior to executing a transaction.

> **Note:** For command–level programs, this user exit must end in the letter "X": GTB@SCTX. Macro–level programs must end in the letter "M": GTB@SCTM.
>
> Also, remember that if you are using CICS version 3.2 and above, you must code the exit in command–level code. If you are using CICS version 3.1 and below, you can code the exit in either command–level or macro–level code.

Passing Parameters

For internal security user exits written in command–level code, BMS/TS uses the COMMAREA to pass control and specified parameters to the user routine. The COMMAREA also contains return code fields to notify BMS/TS, upon return from the exit, what action to take.

See User Exit Parameters for a description of all fields that can appear in the COMMAREA.

GT$UTYPC Return Field

BMS/TS provides a return field (GT$UTYPC) to indicate to the exit the type of BMS/TS action that is currently taking place. BMS/TS passes the GT$UTYPC field to the user routine.

GT$UTYPC Value	Description
c'M'	User exit called from BMS/TS map.
c'U'	User exit called from BMS/TS menu.
c'W'	User exit called from BMS/TS window.

GT$URETC Return Field

BMS/TS provides a return field (GT$URETC) that lets the exit deny or allow a validation request. GT$URETC is passed from BMS/TS to the user routine, and/or from the user routine to BMS/TS.

GT$URETC Value	Description
x'00'	Allows the user exit to skip all validation steps.
x'04'	Allows the user exit to validate the operator ID as previously defined in BMS/TS.
x'FF'	Allows the exit to cancel the transaction.

BMS/TS Programs and User Exit for Internal Security

You can link–edit the following BMS/TS programs to call the user exit for internal security.

Program	Description
GTB$DEL	Member delete
GTBHELP	General help processor
GTB$TRCE	Internal trace
GTBHMAP	Add/update help maps
GTBDDEM	DLIST *DEMO members
GTBGMEN	General menu processor
GTBDJCL	DLIST *JCL members
GTBHUPD	Help add/update processor
GTBDLIS	General DLIST program
GTBIUTL	Copy/delete/rename utility
GTBDMAP	Display *MAPS members
GTBJDEF	Add/update *CJCL
GTBDOPB	Display *OPID members
GTBMAIN	Main entry control processor
GTBDSMO	Display *SYSTEM members
GTBMDEF	*CMAPS member name input program
GTBDTAB	Display *TABLE members
GTBMDIS	General member display program
GTBEDEM	Add/update *DEMO members

Program	Description
GTBEMAP	*MAPS member name input program
GTBMUTL	Administrative utility menu
GTBEOPD	*OPID member name input program
GTBNFLD	Add/update *IFIELD members
GTBESMO	*SYSTEM member name input program
GTBNMAP	Add/update internal maps
GTBETAB	*TABLE member name input program
GTBNMEN	Add/update *IMENU members
GTBNWND	Add/update *IWINDOW members
GTBTUIO	*TABLE online interface program
GTBRETI	Retrieve member name input program
GTBUHLP	Main help control
GTBRETR	File retrieve/copy/move fetch
GTBUUTL	Copy/rename/delete utility

GTBMERG	System monitor table merge
GTBEJCL	Add/update *JCL members
GTBMFED	Display *MAPS program

GTBSIGN	Application interface sign–on program
GTBVAL1	Input rules stage 1 processor
GTBSUBI	*MAPS submits command processor
GTBVAL2	Input rules editor maintenance
GTBTDIS	*TABLE display program
GTBVALR	Main input rules interface
GTBTSUB	*TABLE submit command processor
GTBWNDO	Window help maintenance program

COMMAREA Examples

The following example shows the COMMAREA used by a COBOL exit.

```
LINKAGE SECTION
*     COMMON PASSED AREA
01    DFHCOMMAREA.
      05 GT$-UXCLN      PIC S9(4) COMP.
      05 GT$-URETC      PIC X.
      05 GT$-UXTRN      PIC X(4).
      05 GT$-UXCMD      PIC X(8).
      05 GT$-UXSET      PIC X(8).
      05 GT$-UTYPC      PIC X.
```

The following figure shows the COMMAREA used by an Assembler exit.

```
*         COMMON PASSED AREA
GT$UEXIT  DS     OF     COMMON AREA PASSED BY GTS
GT$UXCLN  DS     H      LENGTH OF USER EXIT A
GT$URETC  DS     C      RETURN CODE
*                         X'00'ACCEPT-NO SECURITY
*                         X'04'NORMAL BMS/TS SECURITY
*                         X'FF'REJECT
GT$UXTRN  DS     CL4    USER TRAN CODE REQUEST
GT$UXCMD  DS     CL8    COMMAND
GT$UXSET  DS     CL8    SET/MEMBER NAME
GT$UMAP   DS     CL8    USED AS *FIELD INTERNAL
GT$UTYPC  DS     C      MEMBER CODE
*                         C'M' *MAPS
*                         C'U' *MENUS
*                         C'W' *WINDOWS
```

Include Modules

Select the appropriate include module(s) for use with the internal security user exit.

Include modules are contained in the following linkset members in your object library: GTB#LNK, GTB$LNK, or GTBLNK.

> **Note:** The following include modules are shown for OS/390 systems. For VSE, delete the references to the library (SYSLIB or CICSLIB), remove parentheses from the member names, and omit the NAME statements.

Refer to GTB@SCTt Include Modules for sample control statements for linking BMS/TS load modules.

Writing GTB@SCTX, Command–Level User Exit for Internal Security

Perform the following steps to write the command–level user exit for internal security.

1. Code the GTB@SCTX exit routine as a CICS program. Remember, the standard COMMAREA transfers information to and from the user exit.

   Note: The name of the user exit stub ends in a "C": GTB@SCTC.

2. Assemble the program, making certain to include the COMMAREA.

3. Create links to pass control from BMS/TS to your own program. To do this, link–edit GTB@SCTX as a standalone load module in CICS. Be sure to create a CSD entry for this program.

4. Re–link the desired BMS/TS calling program. Also, remember to include a statement for GTB@SCTC.

   Note: A list of BMS/TS programs that you can use with the Internal Security user exit, as well as the sample control statements for linking BMS/TS load modules, appears in GTB@SCTt Include Modules.

Writing GTB@SCTM, Macro–Level User Exit for Internal Security

Perform the following steps to write the macro–level user exit for internal security.

1. Code the GTB@SCTM exit routine as a CICS program. Use the GTB@SCTX (command–level exit) sample as a guide, but be sure to use macro–level constructs.

   Note: When using macro–level exits with the Assembler language, CICS accesses data through register 1, which points to the standard parameter list. The first address in this list points to the pass area.

2. Assemble the program.

3. Create links to pass control from BMS/TS to your own program. To do this, link your program as part of the total GT load module with which the program is associated.

4. Re−link the desired BMS/TS calling program. Also, remember to include a statement for GTB@SCTM. You should assemble and place your user exit in your BMS/TS (or other) OBJLIB.

   Note: A list of BMS/TS programs that you can use with the Internal Security user exit, as well as the sample control statements for linking BMS/TS load modules, appears in GTB@SCTt Include Modules.

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

GTB@SCTt Include Modules

Select the appropriate include module(s) for use with the Internal Security user exit (GTB@SCTt).

Include modules are contained in the following linkset members in your object library: GTB#LNK, GTB$LNK, or GTBLNK.

> **Note:** The following include modules are shown for OS/390 systems. For VSE, delete the references to the library (SYSLIB or CICSLIB), remove parentheses from the member names, and omit the NAME statements.

Include Modules

Select from the following list to view sample control statements for linking BMS/TS load modules.

GTB$DEL	GTBJDEF	GTBEMAP	GTBUHLP
GTBHELP	GTBDOPD	GTBMUTL	GTBRETR
GTB$TRCE	GTBMAIN	GTBEOPD	GTBUUTL
GTBHMAP	GTBDSMO	GTBNFLD	GTBSIGN
GTBDDEM	GTBMDEF	GTBESMO	GTBVAL1
GTBGMEN	GTBDTAB	GTBNMAP	GTBSUBI
GTBDJCL	GTBMDIS	GTBETAB	GTBVAL2
GTBHUPD	GTBEDEM	GTBNMEN	GTBTDIS
GTBDLIS	GTBMERG	GTBNWND	GTBVALR
GTBIUTL	GTBEJCL	GTBTUIO	GTBTSUB
GTBDMAP	GTBMFED	GTBRETI	GTBWNDO

Include Modules GTB$DEL

Use the following control statements to re–link the GTB$DEL module (the BMS/TS member delete function).

```
INCLUDE SYSLIB(GTB$DEL)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB$$ERG)
```

```
INCLUDE SYSLIB(GTB$$MOR)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
NAME    GTB$DEL(R)
```

Include Modules GTB$TRCE

Use the following control statements to re-link the GTB$TRCE module (the BMS/TS internal trace function).

```
INCLUDE CICSLIB(DFHEAI)
INCLUDE SYSLIB(GTB$TRCE)
INCLUDE CICSLIB(DFHEAIO)
INCLUDE SYSLIB(GTB$$IM1)
INCLUDE SYSLIB(GTB$$SGT)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$IM2)
INCLUDE SYSLIB(GTB$$PPN)
INCLUDE SYSLIB(GTB$$COS)
INCLUDE SYSLIB(GTB$$EXP)
INCLUDE SYSLIB(GTB$$OOS)
INCLUDE SYSLIB(GTB$$TOS)
INCLUDE SYSLIB(GTB$$SCT)
INCLUDE SYSLIB(GTB$$MOR)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
NAME    GTB$TRCE(R)
```

Include Modules GTBDDEM

Use the following control statements to re-link the GTBDDEM module (the BMS/TS directory list of *DEMO members function).

```
INCLUDE SYSLIB(GTBDDEM)
INCLUDE SYSLIB(GTB$$CPF)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$MOR)
NAME GTBDDEM(R)
```

Include Modules GTBTDIS

Use the following control statements to re-link the GTBTDIS module (the BMS/TS *TABLE display function).

```
INCLUDE SYSLIB(GTBTDIS)
INCLUDE SYSLIB(GTB$$CPF)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$HEXM)
```

```
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$MOR)
NAME    GTBTDIS(R)
```

Include Modules GTBDJCL

Use the following control statements to re−link the GTBDJCL module (the BMS/TS directory list of *DEMO members function).

```
INCLUDE SYSLIB(GTBDJCL)
INCLUDE SYSLIB(GTB$$CPF)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$MOR)
NAME    GTBDJCL(R)
```

Include Modules GTBDLIS

Use the following control statements to re−link the GTBDLIS module (the BMS/TS general directory list function).

```
INCLUDE SYSLIB(GTBDLIS)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$$TYP)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$MOR)
INCLUDE SYSLIB(GTB$$CPF)
NAME    GTBDLIS(R)
```

Include Modules GTBDMAP

Use the following control statements to re−link the GTBDMAP module (the BMS/TS directory list of *MAP members function).

```
INCLUDE SYSLIB(GTBDMAP)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$MOR)
INCLUDE SYSLIB(GTB$$CPF)
NAME    GTBDMAP(R)
```

Include Modules GTBDOPD

Use the following control statements to re−link the GTBDOPD module (the BMS/TS directory list of *OPID members function).

```
INCLUDE SYSLIB(GTBDOPD)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$MOR)
INCLUDE SYSLIB(GTB$$CPF)
NAME    GTBDOPD(R)
```

Include Modules GTBDSMO

Use the following control statements to re−link the GTBDSMO module (the BMS/TS directory list of *SYSTEM MONITOR members function).

```
INCLUDE SYSLIB(GTBDSMO)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$MOR)
INCLUDE SYSLIB(GTB$$CPF)
NAME    GTBDSMO(R)
```

Include Modules GTBDTAB

Use the following control statements to re−link the GTBDTAB module (the BMS/TS directory list of *TABLE members function).

```
INCLUDE SYSLIB(GTBDTAB)
INCLUDE SYSLIB(GTB$$CPF)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$MOR)
NAME    GTBDTAB(R)
```

Include Modules GTBEDEM

Use the following control statements to re−link the GTBEDEM module (the BMS/TS add/update *DEMO members function).

```
INCLUDE SYSLIB(GTBEDEM)
INCLUDE SYSLIB(GTB$$CPF)
INCLUDE SYSLIB(GTB$$HDR)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$MOR)
NAME    GTBEDEM(R)
```

Include Modules GTBEJCL

Use the following control statements to re−link the GTBEJCL module (the BMS/TS add/update *JCL members function).

```
INCLUDE SYSLIB(GTBEJCL)
INCLUDE SYSLIB(GTB$$CPF)
INCLUDE SYSLIB(GTB$$HDR)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$MOR)
NAME    GTBEJCL(R)
```

Include Modules GTBEMAP

Use the following control statements to re−link the GTBEMAP module (the BMS/TS *MAPS member name input function).

```
INCLUDE SYSLIB(GTBEMAP)
INCLUDE SYSLIB(GTB$$CPF)
INCLUDE SYSLIB(GTB$$HDR)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$MOR)
NAME    GTBEMAP(R)
```

Include Modules GTBEOPD

Use the following control statements to re−link the GTBEOPD module (the BMS/TS *OPID member name input function).

```
INCLUDE SYSLIB(GTBEOPD)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$$IFS)
```

```
INCLUDE SYSLIB(GTB$$HDR)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$MOR)
INCLUDE SYSLIB(GTB$$CPF)
NAME    GTBEOPD(R)
```

Include Modules GTBESMO

Use the following control statements to re−link the GTBESMO module (the BMS/TS *SYSTEM Monitor member input function).

```
INCLUDE SYSLIB(GTBESMO)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$$HDR)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$MOR)
INCLUDE SYSLIB(GTB$$CPF)
NAME    GTBESMO(R)
```

Include Modules GTBETAB

Use the following control statements to re−link the GTBETAB module (the BMS/TS *TABLE member name input function).

```
INCLUDE SYSLIB(GTBETAB)
INCLUDE SYSLIB(GTB$$CPF)
INCLUDE SYSLIB(GTB$$HDR)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$MOR)
NAME    GTBETAB(R)
```

Include Modules GTBGMEN

Use the following control statements to re−link the GTBGMEN module (the BMS/TS general menu processor).

```
INCLUDE SYSLIB(GTBGMEN)
INCLUDE SYSLIB(GTB$$MEN)
INCLUDE SYSLIB(GTB$$RMB)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$MENU)
INCLUDE SYSLIB(GTBGMES)
```

```
INCLUDE SYSLIB(GTB$$CTE)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$$MOR)
INCLUDE SYSLIB(GTB$$RED)
INCLUDE SYSLIB(GTB$ESBD)
INCLUDE SYSLIB(GTB$LOCD)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTBCOLEX)
NAME    GTBGMEN(R)
```

Include Modules GTBHELP

Use the following control statements to re−link the GTBHELP module (the BMS/TS general help processor).

```
INCLUDE CICSLIB(DFHEAI)
INCLUDE SYSLIB(GTBHELP)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$$SBF)
INCLUDE SYSLIB(GTB$$FCL)
INCLUDE SYSLIB(GTB$$RHN)
INCLUDE SYSLIB(GTB$$IM1)
INCLUDE SYSLIB(GTB$$COs) * For CICS 3.1 and above, use GTB$$CES
INCLUDE SYSLIB(GTB$$EXP)
INCLUDE SYSLIB(GTB$$OOS) * For CICS 3.1 and above, use GTB$$OES
INCLUDE SYSLIB(GTB$$TOS) * For CICS 3.1 and above, use GTB$$TES
INCLUDE SYSLIB(GTB$$IM2)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$RED)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$MOR)
INCLUDE CICSLIB(DFHEAI0)
NAME    GTBHELP(R)
```

Include Modules GTBHMAP

Use the following control statements to re−link the GTBHMAP module (the BMS/TS add/update help maps function).

```
INCLUDE SYSLIB(GTBHMAP)
INCLUDE SYSLIB(GTB$$CPF)
INCLUDE SYSLIB(GTB$$HDR)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$MOR)
NAME    GTBHMAP(R)
```

Include Modules GTBHUPD

Use the following control statements to re−link the GTBHUPD module (the BMS/TS add/update help processor).

```
INCLUDE SYSLIB(GTBHUPD)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub − use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$MOR)
NAME    GTBHUPD(R)
```

Include Modules GTBIUTL

Use the following control statements to re−link the GTBIUTL module (the BMS/TS copy/delete/rename utility).

```
INCLUDE SYSLIB(GTBIUTL)
INCLUDE SYSLIB(GTB$$CPF)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$$TYP)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub − use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$MOR)
NAME    GTBIUTL(R)
```

Include Modules GTBJDEF

Use the following control statements to re−link the GTBJDEF module (the BMS/TS add/update *CJCL members function).

```
INCLUDE SYSLIB(GTBJDEF)
INCLUDE SYSLIB(GTB$$CPF)
INCLUDE SYSLIB(GTB$$HDR)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub − use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$MOR)
NAME    GTBJDEF(R)
```

Include Modules GTBMAIN

Use the following control statements to re−link the GTBMAIN module (the BMS/TS main entry control processor).

```
INCLUDE CICSLIB(DFHEAI)
INCLUDE SYSLIB(GTBMAIN)
INCLUDE CICSLIB(DFHEAI0)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTBCOLEX)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$IM1)
INCLUDE SYSLIB(GTB$$MEN)
INCLUDE SYSLIB(GTB$$MOR)
INCLUDE SYSLIB(GTB$$MST)
INCLUDE SYSLIB(GTB$$RMB)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$MENU)
INCLUDE SYSLIB(GTB$$CTE)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$$IM2)
INCLUDE SYSLIB(GTB$$RED)
INCLUDE SYSLIB(GTB$ESBD)
INCLUDE SYSLIB(GTB$LOCD)
INCLUDE SYSLIB(GTB$$COs) * For CICS 3.1 and above, use GTB$$CES
INCLUDE SYSLIB(GTB$$EXP)
INCLUDE SYSLIB(GTB$$OOS) * For CICS 3.1 and above, use GTB$$OES
INCLUDE SYSLIB(GTB$$TOS) * For CICS 3.1 and above, use GTB$$TES
NAME    GTBMAIN(R)
```

Include Modules GTBMDEF

Use the following control statements to re−link the GTBMDEF module (the BMS/TS *CMAPS member name input function).

```
INCLUDE SYSLIB(GTBMDEF)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$$HDR)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$MOR)
INCLUDE SYSLIB(GTB$$CPF)
NAME    GTBMDEF(R)
```

Include Modules GTBMDIS

Use the following control statements to re−link the GTBMDIS module (the BMS/TS general member display function).

```
INCLUDE SYSLIB(GTBMDIS)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$$TYP)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$MOR)
```

```
NAME    GTBMDIS(R)
```

Include Modules GTBMERG

Use the following control statements to re−link the GTBMERG module (the BMS/TS System Monitor table merge function).

```
INCLUDE SYSLIB(GTBMERG)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$MOR)
NAME    GTBMERG(R)
```

Include Modules GTBMFED

Use the following control statements to re−link the GTBMFED module (the BMS/TS display *MAPS function).

```
INCLUDE SYSLIB(GTBMFED)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$$TYP)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$MOR)
INCLUDE SYSLIB(GTB$$CPF)
INCLUDE SYSLIB(GTB$$HDR)
NAME    GTBMFED(R)
```

Include Modules GTBMUTL

Use the following control statements to re−link the GTBMUTL module (the BMS/TS administrative utility menu).

```
INCLUDE SYSLIB(GTBMUTL)
INCLUDE SYSLIB(GTBCOLEX)
INCLUDE SYSLIB(GTB$$EI1)
INCLUDE SYSLIB(GTB$$CPF)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$$MEN)
INCLUDE SYSLIB(GTB$$RMB)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$HDLD)
INCLUDE SYSLIB(GTB$MENU)
INCLUDE SYSLIB(GTB$PBCK)
INCLUDE SYSLIB(GTB$SCTY)
```

```
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$CTE)
INCLUDE SYSLIB(GTB$$MOR)
INCLUDE SYSLIB(GTB$$PCL)
INCLUDE SYSLIB(GTB$$PPL)
INCLUDE SYSLIB(GTB$$RED)
INCLUDE SYSLIB(GTB$ESBD)
INCLUDE SYSLIB(GTB$LOCD)
NAME    GTBMUTL(R)
```

Include Modules GTBNFLD

Use the following control statements to re−link the GTBNFLD module (the BMS/TS add/update *IFIELD members function).

```
INCLUDE SYSLIB(GTBNFLD)
INCLUDE SYSLIB(GTB$$CPF)
INCLUDE SYSLIB(GTB$$HDR)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$MOR)
NAME    GTBNFLD(R)
```

Include Modules GTBNMAP

Use the following control statements to re−link the GTBNMAP module (the BMS/TS add/update internal maps function).

```
INCLUDE SYSLIB(GTBNMAP)
INCLUDE SYSLIB(GTB$$CPF)
INCLUDE SYSLIB(GTB$$HDR)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$MOR)
NAME    GTBNMAP(R)
```

Include Modules GTBNMEN

Use the following control statements to re−link the GTBNMEN module (the BMS/TS add/update *IMENU members function).

```
INCLUDE SYSLIB(GTBNMEN)
INCLUDE SYSLIB(GTB$$CPF)
INCLUDE SYSLIB(GTB$$HDR)
INCLUDE SYSLIB(GTB$$IFS)
```

```
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$MOR)
NAME    GTBNMEN(R)
```

Include Modules GTBNWND

Use the following control statements to re−link the GTBNWND module (the BMS/TS add/update *IWINDOW members function).

```
INCLUDE SYSLIB(GTBNWND)
INCLUDE SYSLIB(GTB$$CPF)
INCLUDE SYSLIB(GTB$$HDR)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$MOR)
NAME    GTBNWND(R)
```

Include Modules GTBRETI

Use the following control statements to re−link the GTBRETI module (the BMS/TS retrieve member name input function).

```
INCLUDE SYSLIB(GTBRETI)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$$TYP)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$MOR)
INCLUDE SYSLIB(GTB$$CPF)
NAME    GTBRETI(R)
```

Include Modules GTBRETR

Use the following control statements to re−link the GTBRETR module (the BMS/TS file retrieve, copy, move, fetch function).

```
INCLUDE SYSLIB(GTBRETR)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$MOR)
```

```
NAME    GTBRETR(R)
```

Include Modules GTBSIGN

Use the following control statements to re−link the GTBSIGN module (the BMS/TS application interface sign−on function).

```
INCLUDE SYSLIB(GTBSIGN)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$MOR)
INCLUDE SYSLIB(GTB$$CPF)
NAME    GTBSIGN(R)
```

Include Modules GTBSUBI

Use the following control statements to re−link the GTBSUBI module (the BMS/TS *MAPS submit command processor).

```
INCLUDE SYSLIB(GTBSUBI)
INCLUDE SYSLIB(GTB$$CPF)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$$TYP)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$MOR)
NAME    GTBSUBI(R)
```

Include Modules GTBTSUB

Use the following control statements to re−link the GTBTSUB module (the BMS/TS *TABLE submit command processor).

```
INCLUDE SYSLIB(GTBTSUB)
INCLUDE SYSLIB(GTB$$CPF)
INCLUDE SYSLIB(GTB$$HDR)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$MOR)
NAME    GTBTSUB(R)
```

Include Modules GTBTUIO

Use the following control statements to re−link the GTBTUIO module (the BMS/TS online interface program).

```
INCLUDE CICSLIB(DFHEAI)
INCLUDE SYSLIB(GTBTUIO)
INCLUDE CICSLIB(DFHEAI0)
INCLUDE SYSLIB(GTB$$IM1)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$$IM2)
INCLUDE SYSLIB(GTB$$COs) * For CICS 3.1 and above, use GTB$$CES
INCLUDE SYSLIB(GTB$$EXP)
INCLUDE SYSLIB(GTB$$OOS) * For CICS 3.1 and above, use GTB$$OES
INCLUDE SYSLIB(GTB$$TOS) * For CICS 3.1 and above, use GTB$$TES
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$MOR)
NAME    GTBTUIO(R)
```

Include Modules GTBUHLP

Use the following control statements to re−link the GTBUHLP module (the BMS/TS main help control function).

```
INCLUDE CICSLIB(DFHEAI)
INCLUDE SYSLIB(GTBUHLP)
INCLUDE CICSLIB(DFHEAI0)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$PCTM)
INCLUDE SYSLIB(GTB$PBCK)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$FCL)
INCLUDE SYSLIB(GTB$$IM1)
INCLUDE SYSLIB(GTB$$MOR)
INCLUDE SYSLIB(GTB$$PCL)
INCLUDE SYSLIB(GTB$$PFL)
INCLUDE SYSLIB(GTB$$PPL)
INCLUDE SYSLIB(GTB$$RHN)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$HDLD)
INCLUDE SYSLIB(GTB$$IM2)
INCLUDE SYSLIB(GTB$$RED)
INCLUDE SYSLIB(GTB$$SBF)
INCLUDE SYSLIB(GTB$$COs) * For CICS 3.1 and above, use GTB$$CES
INCLUDE SYSLIB(GTB$$EXP)
INCLUDE SYSLIB(GTB$$OOS) * For CICS 3.1 and above, use GTB$$OES
INCLUDE SYSLIB(GTB$$TOS) * For CICS 3.1 and above, use GTB$$TES
NAME    GTBUHLP(R)
```

Include Modules GTBUUTL

Use the following control statements to re−link the GTBUUTL module (the BMS/TS copy/rename/delete function).

```
INCLUDE SYSLIB(GTBUUTL)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$$TYP)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$MOR)
INCLUDE SYSLIB(GTB$$CPF)
NAME    GTBUUTL(R)
```

Include Modules GTBVAL1

Use the following control statements to re−link the GTBVALI module (the BMS/TS input rules stage 1 processor).

```
INCLUDE SYSLIB(GTBVAL1)
INCLUDE SYSLIB(GTB$$CPF)
INCLUDE SYSLIB(GTB$$HDR)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$MOR)
NAME    GTBVAL1(R)
```

Include Modules GTBVAL2

Use the following control statements to re−link the GTBVAL2 module (the BMS/TS input rules editor maintenance).

```
INCLUDE SYSLIB(GTBVAL2)
INCLUDE SYSLIB(GTBCOLEX)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$HDR)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$EDIT)
INCLUDE SYSLIB(GTB$$CAS)
INCLUDE SYSLIB(GTB$$CTE)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$ESBD)
INCLUDE SYSLIB(GTB$LOCD)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$MOR)
NAME    GTBVAL2(R)
```

Include Modules GTBVALR

Use the following control statements to re−link the GTBVALR module (the BMS/TS main input rules interface).

```
INCLUDE CICSLIB(DFHEAI)
INCLUDE SYSLIB(GTBVALR)
INCLUDE CICSLIB(DFHEAI0)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$$EXP)
INCLUDE SYSLIB(GTB$$IM1)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTB$$VSR)
INCLUDE SYSLIB(GTB$RBLD)
INCLUDE SYSLIB(GTB$$IM2)
INCLUDE SYSLIB(GTB$$COs) * For CICS 3.1 and above, use GTB$$CES
INCLUDE SYSLIB(GTB$$OOS) * For CICS 3.1 and above, use GTB$$OES
INCLUDE SYSLIB(GTB$$TOS) * For CICS 3.1 and above, use GTB$$TES
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$MOR)
NAME    GTBVALR(R)
```

Include Modules GTBWNDO

Use the following control statements to re−link the GTBWNDO module (the BMS/TS window help maintenance function).

```
INCLUDE SYSLIB(GTBWNDO)
INCLUDE SYSLIB(GTB$$SRG)
INCLUDE SYSLIB(GTBWNDE)
INCLUDE SYSLIB(GTB$$DEF)
INCLUDE SYSLIB(GTB$$HTE)
INCLUDE SYSLIB(GTB$$HTA)
INCLUDE SYSLIB(GTB$EDIT)
INCLUDE SYSLIB(GTB$$IFS)
INCLUDE SYSLIB(GTB$$CTE)
INCLUDE SYSLIB(GTB$$CTH)
INCLUDE SYSLIB(GTBCOLEX)
INCLUDE SYSLIB(GTB$$CAS)
INCLUDE SYSLIB(GTB$ESBD)
INCLUDE SYSLIB(GTB$GSBA)
INCLUDE SYSLIB(GTB$$ERG)
INCLUDE SYSLIB(GTB$LOCD)
INCLUDE SYSLIB(GTB$SCTY)
INCLUDE SYSLIB(GTB@SCTC) * User exit stub - use GTB@SCTM for macro code
INCLUDE SYSLIB(GTB$$MOR)
NAME    GTBWNDO(R)
```

Top of page

Contents:

Related topics:

- Customizing with User Exits

Administrator's Guide Home
BMS/TS Doc. Home

GTB@SUBX, User Exit for Job Submission

During this exit, you can perform any CICS command–level function except for screen I/Os or functions which would release any resources.

> **Note:** This exit only supports command–level code.

The user exit for job submission enables you to examine a job prior to its submission so that a user (especially in ACT environments) can allow or reject submission based on certain company–standard criteria. BMS/TS calls this exit for each JCL statement to be processed.

You can use this exit for a variety of purposes:

- Tracking how often data sets are referenced,
- Accepting or rejecting submission of maps to CICS based on user ID, job name, or job class,
- Changing the load library the map is submitted to based on operator ID,
- Checking JCL syntax,
- Resolving symbols (for example, looking for ##OPID1 and replacing it with ##OPID2).

> **Note:** Although you can cancel a job at anytime, make the cancel decision on the first card whenever possible.
>
> If you wish to modify a card, make your changes to the card passed to GTB@SUBX. No special return codes are required.

Passing Parameters

For job submission user exits written in command–level code, BMS/TS uses the COMMAREA to pass control and specified parameters about the job to the user routine. The COMMAREA also contains return code fields to notify BMS/TS, upon return from the exit, what action to take.

See User Exit Parameters for a description of all fields that can appear in the COMMAREA.

Writing GTB@SUBX, Command−Level User Exit for Job Submission

Perform the following steps to write the command−level user exit for job submission.

1. Code the GTB@SUBX exit routine as a CICS program. Remember, the standard COMMAREA transfers information to and from the user exit.

   Note: The name of the user exit stub ends in a "C": GTB@SUBC.

2. Assemble the program, making certain to include the COMMAREA.

3. Create links to pass control from BMS/TS to your own program. To do this, link−edit GTB@SUBX as a standalone load module in CICS. Be sure to create a CSD entry for this program.

4. Re−link the BMS/TS calling program. Also, remember to include a statement for GTB@SUBC.

GTB@SUBX Example

A sample Assembler program appears below. Keep in mind that this is only an example: your program must be specific to the files and libraries used at your company.

```
*************************************************************
* SAMPLE SUBMIT EXIT COMMAND LEVEL CONVENTIONS
* THIS SAMPLE WILL CHANGE THE
* JCL STATEMENT * EXAMPLE TO
* * SUCCESS.
*************************************************************

         TITLE 'GT@SUB - USER INTERFACE FOR BMS/TS '

WORKAREA DSECT
         DS H LENGTH OF THE IS AREA
RTRNCODE DS X RETURN CODE
* PASSED BACK
* X'00' NORMAL RETURN
* > 0 REJECT AND CANCEL
         DS XL5 ** RESERVED **
CARD DS CL80 JCL STATEMENT
*
WORK2 EQU 2
GTB@SUBX CSECT ,
         L WORK2,DFHEICAP PICKUP ADDR OF PARMS
         USING WORKAREA,WORK2 SETUP BASE FOR COMMAREA
*
         CLC CARD(9),=C'* EXAMPLE' IF EXAMPLE, ALTER IT
         BE ALTER
         CLC CARD(7),=C'* ABORT ' IF ABORT, CANCEL
         BE ABORT
         B OK OTHERWISE, JUST GET OUT
ABORT DS 0H
         MVI RTRNCODE,X'FF' CANCEL
         B EXITPGM
ALTER DS 0H
         MVC CARD(9),=C'* SUCCESS' ALTER STATEMENT
```

```
OK MVI RTRNCODE,X'00' GOOD RETURN CODE
EXITPGM  DS 0H
         EXEC CICS RETURN
*
         END
```

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Transaction Monitor
- Manage Active Tables
- Terminal Compression Statistics
- Turn On/Off Terminal Compression
- Turn On/Off User Trace
- Turn On/Off Debug Trace
- Run Memory Display

Administrator's Guide Home
BMS/TS Doc. Home

Using the System Functions

The BMS/TS system functions are online utility programs designed to speed up and assist the operation and use of your system. The system functions provide access to:

- Transaction Monitor option,
- Display of tables loaded into CICS memory,
- Data Compression facility,
- Debugging aids,
- Memory Display.

These online utilities allow you to identify problems, speed up transmission time and display information about how your system's performance.

Access the System Functions

You access the system functions from the System Functions Menu.

1. From the Main Menu, select the System option, and press Enter. The System Functions Menu displays (view sample).

2. Select the desired function from the menu.

   - Transaction Monitor...
   - Manage Active Tables
   - Terminal Compression Statistics
   - Turn On/Off Terminal Compression
   - Turn On/Off User Trace
   - Turn On/Off Debug Trace
   - Run Memory Display

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Using the System Functions

Administrator's Guide Home
BMS/TS Doc. Home

Transaction Monitor

BMS/TS enables you to create help screens and input rules. You use the Transaction Monitor to implement the help screens and input rules you've created.

The Transaction Monitor:

- Keeps track of all currently running applications,
- Determines which of those applications also include help screens and/or input rules,
- Determines if during input the operator pressed the function key that has a corresponding help screen.

After determining the above information, BMS/TS then reroutes the operator's input so it processes through the input rules, and/or displays the appropriate help screen for the transaction.

Access the Transaction Monitor Options

1. From the Main Menu, select the System option and press Enter. The System Functions Menu displays (view sample).

2. From the System Functions Menu, select the Transaction Monitor option and press Enter. The Transaction Monitor Options menu displays (view sample). The menu options are described below.

Transaction Monitor Option Menu

The Transaction Monitor Option menu provides the following options:

Menu Option	Description
Load Monitor	Load a Transaction Monitor into CICS memory. This activates help screens and input rules.
Edit Monitor	Add or change transaction codes that have corresponding input rules, or assign an individual help screen to a transaction code and function key. You can also update any previous assignments of transaction codes, function keys, and help screens.
Refresh Monitor	Reload a Transaction Monitor into CICS memory to reactivate help screens and input rules.
Unload Monitor	Unload a Transaction Monitor from CICS memory. This disables help screens and input rules and releases any work areas required by the Transaction Monitor.

Merge Monitors	Copy all or part of the entries stored in one Transaction Monitor into another Transaction Monitor.
Suspend Monitor	Temporarily disconnect the Transaction Monitor to deactivate help screens and input rules. This does not release the work areas required by the Transaction Monitor.

Understanding the Transaction Monitor

In the Transaction Monitor table, you associate a transaction with input rules and/or the appropriate help screens and function keys. You store the completed Transaction Monitor in the current library, where it remains inactive until you load it into CICS memory.

> **Note:** You can load the Transaction Monitor automatically at CICS system start–up, if desired.

BMS/TS requires that the Transaction Monitor you load into CICS memory have a special set name of ***SYSTEM** and a member name of **MONITOR**. However, you can create a Transaction Monitor table using whatever naming convention your company requires. However, to activate the Transaction Monitor, you must either:

- Rename your library member (set name *SYSTEM, member name MONITOR), or
- Merge it with an existing Transaction Monitor named *SYSTEM/MONITOR.

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Using the Transaction Monitor

Administrator's Guide Home
BMS/TS Doc. Home

Load the Transaction Monitor

Loading a Transaction Monitor table into CICS memory activates help screens and input rules.

> **Note:** To implement input rules, you must replace your program name in the Program Control Table (PCT) with the BMS/TS program name, GTBTHLP, when you load the Transaction Monitor.

Remember that BMS/TS can only load a Transaction Monitor that has a set name of *SYSTEM and a member name of MONITOR into CICS memory. You can create a monitor table with another name, but before loading you must rename it *SYSTEM/MONITOR or merge it with *SYSTEM/MONITOR.

Load the Transaction Monitor Table

You can load the Transaction Monitor into CICS memory.

1. From the Transaction Monitor Options screen, select the Load Monitor option and press Enter. The Load Monitor screen displays (view sample).

2. Press Enter. BMS/TS loads the Transaction Monitor then displays the message System monitor initialization complete: ON.

   Note: If the Transaction Monitor is already loaded into memory, BMS/TS displays the message System monitor currently active.

3. Press F3. BMS/TS returns to the Transaction Monitor Options menu.

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Edit the Transaction Monitor Table

Editing the Transaction Monitor lets you add transaction codes that have corresponding input rules, or assign an individual help screen to a transaction code and function key. You can also update any existing transaction code entries, function keys and help screens.

Select a Transaction Monitor Table

You can select the Transaction Monitor table that you want to edit.

1. From Transaction Monitor Options screen, select the Edit Monitor option and press Enter. The Select a Transaction Monitor screen displays (view sample). The cursor moves to the Transaction Monitor Name field.

2. Press Tab to accept the default (MONITOR), or type the name of the Transaction Monitor to which you want to add an entry. To display a list of valid Transaction Monitor names, press F2.

   Note: Remember, BMS/TS can only load a Transaction Monitor that has a set name of *SYSTEM and a member name of MONITOR into CICS memory. You can create a monitor table with another name, but before loading you must rename it *SYSTEM/MONITOR or merge it with *SYSTEM/MONITOR.

3. If you password–protected the Transaction Monitor table, type the password in the Password field.

4. Press Enter. BMS/TS displays the Transaction Monitor Control screen (view sample).

Edit the Transaction Monitor

Editing the Transaction Monitor lets you add entries that define the transactions which BMS/TS should monitor for help screens and/or input rules. If necessary, you can also use this function to modify an existing entry in the Transaction Monitor.

After you select a Transaction Monitor table, the Transaction Monitor Control screen appears (view sample).

1. Press Tab to move to any LC (Line Command) field.

2. Type I (Insert) and the number of lines you want to insert.

   Note: BMS/TS automatically sorts all entries when you exit the Transaction Monitor. This means that you can enter your information in any order using a line command to insert blank lines. See Appendix A, Using the Line Editor, for a description of the line commands you can use for this screen.

3. Press Enter. BMS/TS inserts the desired number of lines, then moves the cursor to the Tran field of the first blank line.

4. Type the transaction code –– as it appears on the PCT –– that you want BMS/TS to monitor for input rules and/or help screens.

5. Press Tab to move to the Fkey field and enter the appropriate value:

To monitor for...	Then...
Help Screens	Specify the function key that calls the help screen. When this key is pressed, BMS/TS displays the appropriate help screens for the maps associated with the transaction code you entered in step 4.
Input Rules	Type *VAL. This tells BMS/TS to use the appropriate input rules to verify all screen input from the maps associated with the transaction code you entered in step 4.

   Note: Type *IGN (Ignore) in this field if your application has maps that are not relevant to help screens or input rules –– for example, a page–build map, which could have more than one map member.

6. Press Tab to move to the File field, and type the prefix of the library where the transaction is stored. You can also leave this field blank to accept the default (**GT**).

   Note: The next three steps do not apply to input rules. If you are adding an entry for input rules, skip to step 10.

7. Press Tab to move to the Set/Type field, then type the mapset name of the help screen (if applicable).

8. Press Tab to move to the Member field.

9. Type the map name of the help screen. Or, if you are using a help screen naming convention, enter one of the following variables:

For...	Enter the variable...
Help display related to cursor row position	***ROW**
Help display related to cursor field position	***ROWCOL**
Help display related to mapset name and cursor field position	***ROWSET**

10. Press Tab to move to the Description field, and type any comments you want to add for entry.

    Note: For input rules, you can leave this field blank to accept the default, ***BMS/TS VALIDATION***.

11. Repeat steps 1–10 until you have completed your entries.

12. Press F3. BMS/TS displays the message SORT Processing. BMS/TS then clears the screen and displays a screen offering to load the Transaction Monitor.

13. Press Enter if you want to load the Transaction Monitor now. Or press Clear to exit this screen without loading the Transaction Monitor and return to the Select a Transaction Monitor screen.

    - If you press Enter, BMS/TS loads the Transaction Monitor (which immediately implements the help screens and/or input rules), then returns to the Select a Transaction Monitor screen.

    - If you press Clear, you must reload (refresh) the Transaction Monitor in order to activate your changes.

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Using the Transaction Monitor Table

Administrator's Guide Home
BMS/TS Doc. Home

Refresh the Transaction Monitor Table

Refreshing the Transaction Monitor lets you reload the Transaction Monitor in order to activate any changes you may have made.

Remember, if you add input rule or help screen entries to the Transaction Monitor, you must reload the Transaction Monitor in order to activate those changes.

Reload the Transaction Monitor Table

You can reload the Transaction Monitor to activate your updates.

1. From the Transaction Monitor Options screen, select the Refresh Monitor option and press Enter. The Reload Monitor screen displays (view sample).

2. Press Enter. BMS/TS reloads the Transaction Monitor then displays the message System monitor initialization complete: ON.

3. Press F3. BMS/TS returns to the Transaction Monitor Options menu.

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Using the Transaction Monitor Table

Administrator's Guide Home
BMS/TS Doc. Home

Unload the Transaction Monitor Table

Unloading a Transaction Monitor removes the monitor from CICS memory. This will deactivate all help screens and input rules.

Unload the Transaction Monitor Table

You can unload the Transaction Monitor from CICS memory.

1. From the Transaction Monitor Options screen, select the Unload Monitor option and press Enter. The Unload Monitor screen appears (view sample).
2. Press Enter. BMS/TS unloads the Transaction Monitor then displays the message System monitor exit back out complete.
3. Press F3. BMS/TS returns to the Transaction Monitor Options menu.

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Using the Transaction Monitor Table

Administrator's Guide Home
BMS/TS Doc. Home

Merge Transaction Monitor Tables

You can create Transaction Monitor tables using any naming convention. However, BMS/TS will only load a Transaction Monitor with the set/type *SYSTEM and the member name MONITOR.

You can either rename your table to meet this requirement, or you can merge your table entries into the *SYSTEM/MONITOR table.

Merging Transaction Monitors lets you merge information from separate library files, or from the same library file, into a Transaction Monitor named *SYSTEM/MONITOR.

Merge Transaction Monitor Tables

You can copy all or part of the entries stored in one Transaction Monitor into another Transaction Monitor.

1. From the Transaction Monitor Options screen, select the Merge Monitors option and press Enter. The Merge Transaction Monitor Control Tables screen appears (view sample).

   The cursor moves to the Transaction ID From field, which allows you to define the specific transaction code you want to copy to, or to define the beginning code when copying a range of transactions.

2. Type the desired transaction code. Enter the appropriate transaction code if you only want to copy a specific transaction, or enter the beginning code if you want to copy a group of transactions.

   Note: You can use wildcards to signify a particular range of transaction codes. Use an asterisk to indicate a wildcard. For example, type A*** in this field to copy all transactions beginning with an A from the supplying library into the receiving library.

3. Press Tab to move to the to the Transaction ID To field, where you can specify the transaction code you want to copy, or define the ending code when copying a range of transactions.

4. Type the desired transaction code. Enter the same transaction code you entered in step 3 if you only want to copy a specific transaction, or enter the ending code if you want to copy a group of transactions.

5. Press Tab to move to the Transaction Monitor Table Name From field.

6. Type the member name of the Transaction Monitor that contains the transactions you want to copy.

7. Press Tab to move to the Transaction Monitor Table Name To field.

8. Type the member name of the Transaction Monitor to which you want to copy the transactions, or press Tab to skip this field and accept the default (**MONITOR**). The cursor moves to the Library Name Prefix From field.

9. Type the 2–character prefix of the library containing the transactions you want to copy, or press Tab to accept the default (**GT**). The cursor moves to the Library Name Prefix To field.

10. Type the 2–character prefix of the library to which you want to copy the transactions, or press Tab to accept the default (**GT**). The cursor moves to the Replace Like Transactions field.

11. If you want BMS/TS to replace any transactions in the receiving library with transactions of the same name from the sending library, type Y (Yes). If not, type N (No) or press Tab to accept the default (No).

12. Press Enter. BMS/TS copies the desired transactions, then displays the Transaction Monitor Control screen for the merged Transaction Monitors. You can edit the resulting (merged) table.

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Using the Transaction Monitor Table

Administrator's Guide Home
BMS/TS Doc. Home

Suspend the Transaction Monitor Table

You can temporarily disconnect the Transaction Monitor Table to deactivate help screens and input rules.

> **Note:** This does not release the work areas required by the Transaction Monitor Table.

Suspending the Transaction Monitor Table will temporarily disable all help screens and input rules.

Suspend the Transaction Monitor Table

Perform the following steps to suspend the Transaction Monitor:

1. From the Transaction Monitor Options screen, select the Suspend Monitor option and press Enter. The Suspend Monitor screen appears (view sample).
2. Press Enter. BMS/TS suspends the Transaction Monitor (*SYSTEM/MONITOR) then displays the message System monitor control back out complete.
3. Press F3. BMS/TS returns to the Transaction Monitor Options menu.

You can use the Load Monitor option to reactivate the Transaction Monitor.

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

Manage Active Tables

You can display a list of all tables loaded into memory for processing. This display provides the date the table was created, the date it was loaded, and the last time the table was accessed.

Before changes to a table on file can be used, you must first unload a table from memory, then reload it. The loaded tables display enables you to quickly identify the loaded tables, and if necessary, unload a table while still on the display screen.

Display Loaded Tables

You can view a list of all tables loaded into memory.

1. From the Main Menu, select the System option and press Enter. The System Functions Menu displays (view sample).

2. Select the Manage Active Tables option and press Enter. The Display Loaded Tables screen displays (view sample).

3. Review the displayed information for each currently loaded table:
   - Name of the table,
   - Date the table was created,
   - Date the table was loaded into memory,
   - Date the table was last accessed.

4. Press F3. BMS/TS returns to the System Functions Menu.

Unload a Currently Loaded Table

You can unload a table from memory.

> **Note:** Tables currently loaded in memory are not affected by changes unless you first unload the tables using your program or the Manage Active Tables option.

Perform the following steps to unload a currently loaded table.

1. From the Main Menu, select the System option and press Enter. The System Functions Menu displays (view sample).

2. From the System Functions Menu, select the Manage Active Tables option and press Enter. The Display Loaded Tables screen displays (view sample).

3. Press Tab to move the cursor to the left of the name of the table you want to unload.

4. Type D.

5. Press Enter. BMS/TS unloads the table from memory.

6. Press F3. BMS/TS returns to the System Functions Menu.

BMS/TS will automatically reload the table when it is needed. You do not have to reload the table manually.

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Using the System Functions

Administrator's Guide Home
BMS/TS Doc. Home

Terminal Compression

You can activate or deactivate the BMS/TS 3270 Data Compression facility.

> **Note:** The 3270 Data Compression Utility is available only if your company has licensed it from GT Software.

Turn On/Off Terminal Compression

You can activate or deactivate the 3270 Data Compression utility.

1. From the Main Menu, select the System option and press Enter. The System Functions Menu displays (view sample).

2. From the System Functions Menu, select the Turn On/Off Terminal Compression option and press Enter. The Turn On/Off Compression screen displays (view sample).

3. Perform the desired action.

If you want to...	Then...
Turn on the 3270 Data Compression utility,	Type ON or press F5.
Turn off the 3270 Data Compression utility,	Type OFF or press F6.

- If activating the 3270 Data Compression utility, BMS/TS displays the message $906 3270 compression invoked.
- If deactivating the 3270 Data Compression utility, BMS/TS displays the message $906 3270 compression deactivated.

4. Press F3. BMS/TS returns to the System Functions Menu.

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Using the System Functions

Administrator's Guide Home
BMS/TS Doc. Home

Terminal Compression Statistics

You can display statistics about the BMS/TS terminal compression utility.

> **Note:** The 3270 Data Compression Utility is available only if your company has licensed it from GT Software.

You can analyze the data stream as it is sent to your 3270 terminal. During the analysis the facility searches for and compresses repetitive data (such as binary zeros or blanks) before the data is actually sent to the terminal. This can decrease the amount of transmission time.

This feature also lets you display compression statistics to compare the amount of data sent before and after compression and the time the transmission would have taken with the compression facility enabled or disabled.

Display Compression Statistics

You can compare the total number of uncompressed sends to the total number of sends that have been compressed with the Data Compression utility.

1. From the Main Menu, select the System option and press Enter. The System Functions Menu displays (view sample).

2. Select the Terminal Compression Statistics option and press Enter to display the Display Compression Statistics screen (view sample).

3. Review the displayed information:
   - Total number of data stream transmissions,
   - Total amount of data transmitted (in bytes),
   - Total number of data stream transmissions compressed,
   - Total amount of data compressed (in bytes).

Note: This screen lists compression information for your telecommunications access method,

either basic (BTAM) or virtual (VTAM).

4. Press F3. BMS/TS returns to the System Functions Menu.

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

User Trace

Activate the User Trace to track all help requests in an application you have created with BMS/TS. Before an actual help screen is displayed, this facility displays various information about the help screen, including the transaction code associated with the help screen, mapset and map names, and the function key assigned to the help screen.

> **Note:** You should activate the User Trace only at the request of a GT Software Technical Support Representative.

Turn On/Off User Trace

You can activate or deactivate the User Trace facility.

1. From the Main Menu, select the System option and press Enter. The System Functions Menu displays (view sample).
2. From the System Functions Menu, select the Turn On/Off User Trace option and press Enter. The Turn On/Off User Trace screen displays (view sample).
3. Perform the desired action.

If you want to...	Then...
Turn on the User Trace facility,	Type ON or press F5.
Turn off the User Trace facility,	Type OFF or press F6.

- If activating the User Trace facility, BMS/TS displays the message $902 Software TRACE invoked.
- If deactivating the User Trace facility, BMS/TS displays the message $903 Software TRACE reset.

4. Press F3. BMS/TS returns to the System Functions Menu.

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Using the System Functions

Administrator's Guide Home
BMS/TS Doc. Home

Debug Trace

You can use the Debug Trace option to pinpoint operational problems within BMS/TS. The facility tracks all cycles of the BMS/TS internal programs, then provides information about each requested function for monitoring the flow of programs as they process.

Note: You should activate the Debug Trace only at the request of a GT Software Technical Support Representative.

Turn On/Off Debug Trace

You can activate or deactivate the Debug Trace facility.

1. From the Main Menu, select the System option and press Enter. The System Functions Menu displays (view sample).
2. From the System Functions Menu, select the Turn On/Off Debug Trace option and press Enter. The Turn On/Off Debug Trace screen displays (view sample).
3. Perform the desired action.

If you want to...	Then...
Turn on the Debug Trace facility,	Type ON or press F5.
Turn off the Debug Trace facility,	Type OFF or press F6.

- If activating the Debug Trace facility, BMS/TS displays the message $900 Software DEBUG trace invoked.
- If deactivating the Debug Trace facility, BMS/TS displays the message $901 Software DEBUG trace reset.

4. Press F3. BMS/TS returns to the System Functions Menu.

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Using the System Functions

Administrator's Guide Home
BMS/TS Doc. Home

Memory Display

The Memory Display utility lets you display memory storage areas of any CICS region. You can display information using a storage address, CICS system table, CICS item related table, or a program.

When running the Memory Display utility, BMS/TS displays the CICS Online Memory Display Program screen.

> **Note:** The Memory Display Utility is available only if your company has licensed it from GT Software.
>
> As part of this function, BMS/TS can temporarily load a program into memory for your display. If your system is very active, CICS may replace the program in memory before you are finished with your review. You may want to place the program on hold using the HLD=*pgmname* command.

Run Memory Display

Perform the following steps to run the memory display.

1. From the Main Menu, select the System option and press Enter. The System Functions Menu displays (view sample).

2. From the System Functions Menu, select the Run Memory Display option and press Enter. The Online Memory Display Program screen displays (view sample). The cursor moves to the ENTER ADDRESS field.

3. Perform the desired action.

If you want to...	Then...
View memory in a CICS region by address,	Type the desired storage address in hexadecimal characters.

Access a region by CICS system table,

Type an equal sign (**=**) and the code for the table you want to display. Valid table codes are:

CSA	Common System Area
OPF	Optional feature list
SIT	System Initialization Table
TRT	Trace Table
TST	Temporary Storage Table
TUA	TCTTE User Area

Access CICS item related tables,

Type the appropriate data from the following list:

To view...	Type...
FCT entry	**FCT=*ddname***
PCT entry	**PCT=*tranid***
PPT entry	**PPT=*pgmname***
PCT entry	**PRO=*profile***
TCTTE entry	**TCT=*termid***
Extract exit's global work area	**XIT=*pgmname***
Global work area	**GTM=*termid***

Display a program within CICS,

Type in the name of the program you want to display.

If your system is very active and you want CICS to hold the program in memory until you are finished with your review, type **HLD=*pgmname***.

4. Press Enter. BMS/TS displays the requested storage area.

5. Repeat steps 3 and 4 as needed to display all desired information.

6. Press F3. BMS/TS returns to the System Functions Menu.

You can use the following function keys to help review the displayed information.

Function Key	Description
F1 Help	Displays a description of the codes displayed on screen (address pointers, storage pointers, etc.).
F2 Dump	CICS dump.
F3 Exit	Return to the System Functions Menu.
F7 Backward	Scrolls backward one screen (if applicable).
F8 Forward	Scrolls forward one screen (if applicable).

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

BMS/TS Support for IBM 3270 Bridge

BMS/TS supports the IBM CICS Transaction Server 3270 Web Bridge (IBM 3270 bridge).

The IBM CICS Transaction Server 3270 Web Bridge (introduced in Transaction Server 1.2 and enhanced in TS 1.3) provides a method of accessing existing CICS applications written for 3270 input and output. It enables access to 3270–based CICS applications without the use of 3270 terminals. You can access them from the Web, MQSeries, CICS Business Transaction Services (BTS), or any other non–3270 application.

Refer to the IBM Redbook SG24–5480–00 "CICS Transaction Server for OS/390 Version 1 Release 3: Web Support and 3270 Bridge" for detailed information about the IBM 3270 bridge.

How BMS/TS Works with the IBM 3270 Bridge

BMS/TS support for the IBM 3270 bridge includes building "Application Data Structure" (ADS) in the BMS mapset load module. The ADS is a structure that contains additional information about the 3270 application that is accessing the BMS mapset. In addition to this structure, BMS/TS provides for the base generation of the templates used by the Transaction Server Web Interface.

Support for the the IBM 3270 bridge is based on the URM (User Replaceable Module). However it may not be necessary for you to write anything. CICS TS 1.3 provides three bridge (URM) programs; MQSeries version 2.1 provides an additional exit. One of the supplied exits should support most client needs.

These exits are based on the information supplied via the ADS generated by BMS/TS.

How does BMS/TS support differ from base CICS TS macro support? BMS/TS removed the need to run three different assemble steps to build the required map structures, as these are built directly at map submit time.

Because CICS TS macros do not support line occurs and expanded field occurs, often the ADS structure does not reflect the true description of the data. BMS/TS will use the occurs support defined in the base ADS to allow the definition of line occurs and field occurs, so you have a clearer picture of how the data is used in the target 3270 application.

BMS/TS will support the older forms of HTML template generation as well as the newer forms; there are advantages and disadvantages to both forms. Choose the form that best suits your needs:

- The older form of HTML template generation (before IBM CRI94F) does not support various features (such as multi–send operations and dynamic attribute switching by applications) that may be used in CICS applications. If your application does not use these features, then the newer form is not a requirement. Your control over the final document generated at run time is greater, because dynamic tag generation by the 3270 bridge is not used. This form is suitable for creating documents which are focused toward data requests or situations where the desired output is not a facsimile of the 3270 display. In BMS/TS, we refer to this older form as Type 1.

- The newer form (with the IBM CRI94F updates) allows the 3270 bridge to present HTML that will function more like a terminal interface. This is beneficial if you want to use the 3270 bridge as a gateway for client access to the CICS applications. We refer to this newer form as Type 2.

Configure BMS/TS for Template Generation

Perform these steps to configure BMS/TS for template generation:

- Configure CICS Transaction Server for 3270 Bridge Operation,
- Configure a *JCL Member for CICS Template Submission,
- Configure a GTBUTIL Job for Batch Template Submission.

Configure CICS Transaction Server for 3270 Bridge Operation

You must install and configure the IBM–supplied 3270 Bridge exits in the CICS region. Refer to the appropriate IBM documentation for details about 3270 Bridge exit definition.

Keep in mind:

- Allocate a DFHHTML PDS, if it will be used. Remember to update the startup JCL for the CICS region to include a DD statement referring to this dataset.

- Configure the TCPIPSERVICEs which will provide the HTTP interfaces for the 3270 bridge.

Configure a *JCL Member for CICS Template Submission

To create the templates to be used by the 3270 bridge, you need to create a SUBMIT member which includes the new @HTML parameter, as shown in the following sample.

```
//IEBUPDTE EXEC    PGM=IEBUPDTE,PARM='NEW'
//SYSPRINT DD  SYSOUT=A
//SYSUT1   DD  DSN=YOUR.HTML.PDS,DISP=SHR
//SYSUT2   DD  DSN=YOUR.HTML.PDS,DISP=SHR
//SYSIN    DD  *
@HTML      3270 BRIDGE HTML INSERTED HERE
/*
```

There are several options that you can use with the @HTML parameter:

BRCOMP Indicates whether you want to compress the HTML code.

- BRCOMP=Y compresses the HTML; this is the default value.
- BRCOMP=N does not compress the HTML.

Note: HTML is generated, like the BMS macros, with each tag on its own line. This creates requirements for large buffer sizes used by the CICS Bridge. To reduce these buffer requirements, use BRCOMP=Y to combine tags on lines where room allows.

BRLOOK Indicates how you want to the HTML screens to display.

- BRLOOK=IBM produces a light gray background similar to that of the BMS macro template generation.
- BRLOOK=GT produces a black background similar to a 3270−style display. This is the default value.

Remember, you can edit the resulting HTML to customize the displays.

BRTYPE Indicates the type of field names.

- BRTYPE=1 represents simple field names as implemented in Transaction Server version 1.
  Type 1 example: &FLDNAME (uses full field name).
- BRTYPE=2 represents the field naming conventions used in Transaction Server version 2 in *rrcccllll* (row, column, length) format.
  Type 2 example: &F010020032_FLDNAME (uses up to 21 characters of name).

If you are unsure which to use, BRTYPE=1 is the only option supported by all versions of Transaction Server. Default is **1**.

If you choose to use any of the options, separate them with commas. For example:

```
@HTML BRCOMP=N,BRLOOK=IBM
```

Configure a GTBUTIL Job for Batch Template Submission

You can generate templates using the GTBUTIL batch utility. This procedure uses the BMS/TS CATAL command in a manner similar to physical (and symbolic) mapset generation. The CATAL command now includes several new options for 3270 bridge support.

A new output DD for GTBUTIL has also been added "GT$PCHH" to hold the generated template output prior to writing it to its final destination. GT$PCHH is identical in definition to the GT$PCHD and GT$PCHS datasets.

Sample JCL for Bridge Template Generation Using GTBUTIL

```
//STEP1OF1 EXEC PGM=GTBUTIL
//SYSLST   DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//*                        ============ PLEASE MODIFY
//*                        ¦¦¦¦¦¦¦¦¦¦¦¦
//STEPLIB  DD DISP=SHR,DSN=your.GTBR8000.TEST.LOADLIB
//GT$FILE  DD DISP=SHR,DSN=your.GTBR800.GT$FILE
```

```
//GT$PCHS  DD SYSOUT=*
//GT$PCHP  DD SYSOUT=*
//GT$PCHH  DD SYSOUT=*
//SYSIN    DD *
    CATAL MAPSET=ACCT,BRIDGE=YES,ADS=YES,SOSI=NO
/*
```

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Bridge Web Services (BWS)
- BWS Server Administration

User's Guide Home
BMS/TS Doc. Home

BWS Analyzer Initialization File

The BWS Analyzer is controlled by the GT:/ETC/BMS_BWS.INI file. This file contains all the information to customize the execution of the Analyzer program.

The text–based file contains several sections and each section contains various control settings, as noted below.

You can use a text editor to manage the INI file, but the administration panels provide an easy way to update and maintain the file.

{DAEMON} This section contains general server options.		
Parameter		***Value/ Description***
server_root	=	CWS:/WWW/ HFS path information
Server_Name	=	BMS/TS BWS Description
Secure_Socket	=	*n* where *n* equals **0** (No) or **1** (Yes)
Server_Addr	=	*n.n.n.n* IP address of the server
Server_Host	=	xxx.xxx.xxx Server host name
Server_Port	=	*nn* Server port number
LUA_Script_Path	=	CWS:\LUA Additional Lua path
Index_Icon	=	<img src=...> HTML image tag for Index Icons
Index_Parent_Icon	=	<img src=...> HTML image tag for Parent Index Icons
Index_Dir_Icon	=	<img src=...> HTML image tag for Directory Index Icons
Index_File_Icon	=	<img src=...>

		HTML image tag for File Index Icons

{INDEX_DOCS}
This section controls what description will be displayed for a file when building a dynamic index html of a directory.

*.xxx	=	Description Character descriptor of the file extension

{INDEX_ICONS}
This section controls which graphic images are used when building a dynamic index html of a directory.

*xxx	=	<img src=...> HTML image tag INDEX ICON for this file type

{DEFAULT_INDEX}
This section controls the order in which files will be selected for the default index in a directory. If the file is found it will be the default index html for the target directory. The special character '*' informs the server to build a dynamic index html of the directory.

1	=	1sttry.html
2	=	2ndtry.html
3	=	* List of HTML files to use for index (*=Dynamic Index)

{LOG}
The LOG section informs the server which CICS TDQ to use for the three reports. The success report lists files that were accessed correctly. The error report lists files that were not found (file missing, or broken links, or user errors). The trace report is used to debug server problems.

success	=	xxxxxxxx TDQ for success log
errors	=	xxxxxxxx TDQ for error log
trace	=	xxxxxxxx TDQ for trace log

{REDIR}
This section allows you to redirect incoming URLs to another HTTP Web site address. The redirect has many different functions; for example, in–house Web designers can code links in their HTML that get redirected when selected. This can reduce the HTML changes that might be required to keep external links current. The '*' character in the path is used as a wildcard character and ends the compare at that point.

/path*	=	http://www.xyz.com Redirect path to new URL

{ALIAS}

This section allows the Web designer to move files or hide the true location of files from the Web client.		
/path*	=	vol:/path Path ALIAS control to redirect URL path to new HFS path

{METHOD} The Method section is the heart of the RULES BASED CICS Web Interface analyzer. The path information in the URL is compared to the path or file extension information to select the correct Service Method Section in this file. The Service Method Section names are user–defined, but they must be unique within the file, or the results will be unpredictable. The Service Method section will define what controls are used when starting the server program.		
/path*	=	*name* where *name* is a user–defined method section in this file

{USER–DEFINED–METHOD–SECTION} The User Defined Service Method Section is used to inform the BWS Analyzer program what functions to perform or not perform. The best part about the BWS Analyzer is that the true methods behind the URL are hidden from the browser user. An additional benefit is that Web designers do not have to code full paths in their HTML.		
ALIAS_TRANID	=	*xxxx* where *xxxx* is a user–defined transaction code for the alias transaction
ALIAS_TERMID	=	*xxxx* where *xxxx* is a user–defined terminal ID for the alias transaction
CONVERSION	=	DFHWBUD DFHWBUD or a URM for doing the code page conversions
CONVERTER	=	*xxxxxxxx* where *xxxxxxxx* is a URM for the conversion program
ESCAPE	=	DFHWBUN DFHWBUN or a URM for doing the escape/unescape characters
CONVERSION	=	DFHWBUD DFHWBUD or a URM for doing the code page conversions
SERVICE	=	xxxxxxxx or * CWS application to call to service the URL. The * will cause the service name to be pulled from the URL
TOKEN	=	*xxxxxxxx* where *xxxxxxxx* is a static token to pass
USERID	=	*xxxxxxxx* or * where *xxxxxxxx* is a static user ID or, when *, use the BASIC–AUTHENICATE user ID

{MIME} The **BMS**/**TS** standard MIME table can be extended or modified using this section. The information entered here will be placed in the Content_Type header.		
xxx	=	xxxx/yyyy

		Override MIME type for this file extension
.shtm	=	text/x–server–parsed–html The x–server is a special setting for Server Side Include members

{AUTHORIZATION} This section defines the type of security used for the server.		
program	=	xxxxxxxx or * URM name or * for BMS/TS Security
/path*	=	basic authentication realm string When the incoming URL path finds a match, the string entered will be used for the basic authentication challenge

{Secured_Users} This section contains a list of users and their passwords for the BWS processing.		
userid	=	{usection}

{users} The user sections contain the password and access permissions for specific users.		
password	=	value The value must be * or a base 64 encrypted password
admin	=	**0** (user does not have Admin permissions) or **1** (user does have Admin permissions)
access	=	**A** (allow) or **D** (disallow) access
path	=	**A** (allow) or **D** (disallow) access to specific path(s)

{GET} ***{POST}*** ***{PUT}*** This section provides a way to secure the server against an incoming IP address.		
n.n.n.n	=	Allow / Disallow Allow or disallow access to the IP address

{MESSAGE} This section provides the names for HTML members that are sent when errors occur in the server.		
nnn	=	vol:/path/file.shtm When the *nnn* message occurs, send the HTML in the path

{SERVICE_DEFAULT} The Service Default section specifies the initial settings for all services that the BWS anaylzer will call. The User Defined Method sections will override these settings.		
ALIAS_TRANID	=	*xxxx* where *xxxx* is a user–defined transaction code for the alias transaction
ALIAS_TERMID	=	

		xxxx where *xxxx* is a user–defined terminal ID for the alias transaction
CONVERSION	=	DFHWBUD DFHWBUD or a URM for doing the code page conversions
CONVERTER	=	*xxxxxxxx* where *xxxxxxxx* is a URM for the conversion program
ESCAPE	=	DFHWBUN DFHWBUN or a URM for doing the escape/unescape characters
CONVERSION	=	DFHWBUD DFHWBUD or a URM for doing the code page conversions
SERVICE	=	*xxxxxxxx* or * CWS application to call to service the URL; the * will cause the service name to be pulled from the URL
TOKEN	=	*xxxxxxxx* where *xxxxxxxx* is a static token to pass
USERID	=	*xxxxxxxx* or * where *xxxxxxxx* is a static user ID, or when * use the BASIC–AUTHENICATE user ID

{HOST}		
n.n.n.n	=	vol:/path Incoming IP address redirection

{INI}		
PROTOCOLS	=	vol:/path Path and filename for the protocols.ini file
SERVICES	=	vol:/path Path and filename for the protocols.ini file
HOSTS	=	vol:/path Path and filename for the hosts.ini file
NETWORKS	=	vol:/path Path and filename for the networks.ini file
RESOLVE	=	vol:/path Path and filename for the resolve.ini file

---

Top of page

User's Guide Home
BMS/TS Doc. Home

BWS Server Administration

The BWS Server Administration Manager enables you to administer or view your server from remote locations using a standard Web browser.

Most daily administration is done using the Server Administration Manager. Other settings can be changed by editing the server initialization files using a text editor.

The administration panels allow you to maintain settings in the following INI files:

- BMS_BWS.INI file,
- GTBIPFTP.INI file,
- GTBIPSRV.INI file.

Access the Administration Screens

To access the BWS Server administration screens, point your browser to <http://*local_host:port*/bws.lua>. You'll see the main BWS administration screen.

The main BWS administration screen offers the following options:

- Server controls,
- Server methods,
- Server security,
- FTP server.

Server Controls

The Server Controls panel allows you to maintain the following sections of the gt:/etc/bms_bws.ini file:

- Daemon section,
- Log section,
- Index section,
- URL Alias section,
- URL Redirect section,
- MIME section.

Server Methods

The Server Methods panel allows you to maintain the Application Method section of the gt:/etc/bms_bws.ini file.

Server Security

The Server Security panel allows you to maintain the following sections of the gt:/etc/bms_bws.ini file:

- Basic Authorization,
- Secured Users.

FTP Server

The FTP Server panel allows you to maintain the FTP–related settings in:

- FTP Daemon section of the gt:/etc/gtbipftp.ini file,
- FTP Process section of the gt:/etc/gtbipsrv.ini file,
- FTP Users section of the gt:/etc/gtbipftp.ini file.

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

FTP Server Administration

The FTP Server is a native CICS application. It processes FTP requests over TCP/IP. To start the FTP Server, enter the transaction GTRV.

This allows the server to provide access to BMS/TS HFS files hosted by the FTP server from PCs equipped with a browser or FTP program. No 3270 emulation is required.

The FTP Server is an optional feature of BMS/TS. You must be licensed to use the FTP Server. The FTP Server provides built–in security via the GTBIPFTP.INI file, and a URM is supplied to allow for RACF security verification.

FTP Server Initialization (INI) Files

The FTP Server uses the GTBIPSRV.INI file for general settings. Authorizations for using FTP on the FTP Server are set in the GTIIPFTP.INI file.

You may also need to change settings in the following files:

- Hosts.ini,
- Resolve.ini.

FTP Server Initialization Files

The FTP Server is configured to use port 21 and will work without modification if no other FTP server is active in the system. If port 21 is already in use, the FTP server will not be able to activate. Review the TCP/IP settings and select a port number not in use on the system. Remember most FTP clients default to port 21 for FTP, so the client settings for port will need to be adjusted as well.

> **Note**: You can use the supplied server administration tools or edit the INI files using the edit function of the Command Line Processor.

GTBIPFTP.INI File

The FTP configuration file consists of the following sections: {Daemon}, {Scope}, {INI}, {*users*}, {*profiles*}, {Authorization}, {TCPIP}, and {FuncLog}.

{Daemon}	
Server_Name	The name defined to the IP address for the machine as defined in the DNS Domain Name Server or the hosts file in the TCP/IP configuration. *This value is*

	not required and is used for documentation only. Example: ftp.server.com
Server_Port	The port to be used for the FTP process. Normal setting is **21** for FTP processing, but this can be any value if port 21 is already used. Note that the port with a value one less than this setting will also be reserved. For example: If you have more than one CICS system using the FTP server, you might set the port to 2110 for the first server, 2120 for the second, etc. (The system will automatically reserve ports 2109, 2119, etc.) Other standards may exist at your installation; contact the TCP/IP network administrator before you select a value for the port.
Server_Root	The default directory for FTP processing. If allowed by security options, users can change the directory using FTP client commands. Normally this would be set to the most commonly used directory for FTP upload and download processing. The format of this field is *volume:\directory1\directory2...* Example: gt:\www\ftparea. With this setting, every FTP client will have access to the ftparea directory under the www directory of the GT: volume.
Environment	Specifies the number of environment variables to allow. The system reserves 4 bytes of space for each variable allowed. (For example, if you set this value to 350, the system reserves 1400 bytes.) The default value is **200**.
Server_Refresh	**Note:** This setting applies only if you are using the GT Software Novation Document Server and you are accessing the BMS/TS GT$FILE with the Novation Document Server. Allows you to disable the automatic refresh command issued upon completion of each FTP task if there has been an update to HFS. If you are running multiple FTP tasks, you may not want to refresh after each task. Valid values are 0 (Off) and 1 (On). The default value is **1** (automatic refresh is on).

{Scope}	
FTPSCOPE	Set to **No** for BMS/TS and ASSIST/TS.

{INI}	
PROTOCOLS	Identifies the path/filename for the **PROTOCOLS.INI** file. The default is **GT:/ETC/PROTOCOLS.INI**.
SERVICES	Identifies the path/filename for the **SERVICES.INI** file. The default is **GT:/ETC/SERVICES.INI**.
HOSTS	Identifies the path/filename for the **HOSTS.INI** file. The default is

	GT:/ETC/HOSTS.INI.
NETWORKS	Identifies the path/filename for the **NETWORKS.INI** file. The default is **GT:/ETC/NETWORKS.INI**.
RESOLVE	Identifies the path/filename for the **RESOLVE.INI** file. The default is **GT:/ETC/RESOLVE.INI**.

{Authorization}	
Program	URM name control. The URM security exit is called only if an FTP USERID section password control is set to URM, and FTPSCOPE=No is set in the {Scope} section.
Extensions	Activates an extension area to the COMMAREA user by the URM that allows for passing additional information to the URM module. This variable is an integer with a value of at least 15. If a value higher than 15 is entered, the URM will have the ability to use this area as a work area, but the first 15 bytes are reserved. If you use EXTENSIONS, the IP address of the user is added in dotted IP format to the first 15 bytes of the COMMAREA extension area.

{TCPIP}	
Wait	The time, in seconds, of the idle loop. The default value is **60** seconds.
Inbuf	The size of the data buffer (in bytes) of data being sent to the browser. The default is **4096**.
Outbuf	The size of the data buffer (in bytes) of data being sent from the browser. The default is **4096**.

{FuncLog}	
***ftpcommand*=Y**	Call the URM after the execution of each specified command. For example, an entry of STOR=Y will log all STOR commands.

{*users*}	
user*=*profile	Specify the users who have FTP access, and associate each user with a profile that specifies the user's rights (view sample). The *user* name and *profile* name must be one "word", with no embedded spaces. Special characters, such as underscores, are acceptable as long as they will map to a character in code page 500.

{*profiles*}		
Password	***password***	Base 64 encoded password.
	*	No password required.
	URM	Use the program specified in the {Authorization} section to validate the user. The authorization will also be responsible for validating

	FTP commands.
command=value	The FTP commands are listed in the table below. The valid values are: **–1** Disallow command, write to log. **0** Disallow command, no log. **1** Allow command, no log. This is the default value. **2** Allow command, write to log.

FTP Commands

Command	Description
"abor"	Abort current operation
"acct"	ACCT *xxxxxxxx*
"allo"	Allocate space for file
"appe"	Append *xxxxxxx* with data –– UPLOAD APPEND
"cdup"	Change working directory to parent
"cwd"	Change working directory *xxxxxxxx*
"dele"	Delete *xxxxxxx* from file system
"help"	Print help text
"list"	Directory list via data connection
"mkd"	Make directory
"mode"	Mode {s,b,c} –– STREAM, BLOCK, COMPRESSED
"nlst"	Name list via data connection
"noop"	No operation––send positive reply
"pass"	PASS *xxxxxxxx*
"pasv"	Returns port of server for data transfer
"port"	Port h1,h2,h3,h4,p1,p2 of client for data
"pwd"	Print name of current working directory
"quit"	Quit
"rein"	Reinitialize
"retr"	Retrieve *xxxxxxxx* –– DOWNLOAD filename
"rmd"	Remove Directory
"rnfr"	Rename from *xxxxxxxx*
"rnto"	Rename to *xxxxxxxx*
"site"	Site specific options
"smnt"	Structure mount––mount different file system

"stat"	Print FTP server information
"stor"	Store *xxxxxxxx* in file system –– UPLOAD
"stou"	Store unique –– server chooses filename
"stru"	Structure {f,r,p} –– FILE, RECORD, PAGE
"syst"	Print operating system type
"type"	Type {a,e,i,l} –– ASCII, EBCDIC, IMAGE, LOCAL
"user"	USER *xxxxxxxx*
"xpwd"	Print name of current working directory

GTBIPSRV.INI File

The FTP server includes the following sections for configuring the listener (**GTIIPSRV.INI**): {Daemon}, {INI}, {Servers}, {Profile}, and {Administration}.

{Daemon}	
Address	The Internet Address of the server you are defining in dotted decimal format. The default value is **0.0.0.0**, which indicates "bind to any adapter address on this machine".
Timer	The time, in seconds, for the idle loop. At the end of each idle period, the server checks for commands issued through the transient storage queue. The default value is **15** seconds.
Control	Port number used by the controlling program to issue immediate commands to the server. The default value is **69**. *This field is reserved for future use.*

{INI}	
SERVICES	There are five fields in the INI section. However, the only field that you need to set right now is **SERVICES**. (The remaining fields will be used for advanced features in future releases.) The **SERVICES** field identifies the path/filename for the **SERVICES.INI** file. This file associates service names with server ports. The default is **GT:/ETC/SERVICES.INI**.

{Servers}
In this section you define services and their related protocols. The first line is used to add new services. The remaining lines display the existing services, which you can modify, as needed. The services display in the order in which they were defined. To add a service: 1. In the Service/Protocol field, specify the name of the service and the system–level protocol, separated by a slash. (For example, to define FTP service, the entry could be FTP/TCP.) 2. In the Server Profile field, you can add comments following the service entry. 3. When you have completed the entry, select the Add button to save the information. The newly–added service is displayed below the Add field.

{Profile}

There is a PROFILE entry for each defined service. The order corresponds to the order in which the services were defined. In this section, you associate a transaction code and/or program name with each service.

If you specify a transaction, the server will start the transaction when the service is requested. If you specify a program, the server will link to the specified program when the service is requested.

- Enter the CICS transaction code for the service (if any) in the Transid field. The transaction must be defined to CICS. (If there is no Program for the profile, the Transid field is required.)
- Enter the program name (and optional comments) in the Program field. (If there is no Transid for the profile, the Program field is required. If a Transid is entered, the program is ignored.)
- Select the appropriate Change button(s) to update the information. (Remember, if you update both fields, you must select each Change button separately.)

{Administration}

This section identifies the IP addresses, user IDs and passwords that will be accepted for server administration requests. That means the Novation Server will allow administrative requests only from the machines/persons specified here.

The first line is used to add new records. The remaining lines display the existing entries, which you can modify, as needed.

Note: You must initially edit this portion of the INI file manually, to give yourself authorization to perform online service requests. Then any other changes may be performed online. To add an IP address or User/Password:

1. Enter the IP address in the IP Address field. The address must be in dotted decimal format.
2. Specify the Userid:Password, or "*". **Note**: This field is case−sensitive.
    - ♦ To identify a specific user, enter the user ID and password in the specified format. (For example, for user Smith, whose password is "maddog", the entry would be Smith:maddog.)
    - ♦ To allow access to all users from the specified IP address, just type an asterisk (*) in this field.
3. Select the Add button to add the new information.

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- FTP Server Administration

Administrator's Guide Home
BMS/TS Doc. Home

FTP Server: Supplied Security Application

A security application is supplied with the FTP Server.

You can use an external security program to restrict access through the FTP server. You can write the security exit program in any language supported by CICS, and you can even check RACF permissions. Checking RACF allows you to have fewer authorizations files, and so requires less administration time and fewer files.

How Does a Security Program Work?

When the browser requests a file from the server, the server reads the {Authorization} section of the **GTAIPFTP.INI** file. If the requested file is in a secured realm, the server instructs the browser to prompt the user to enter a user ID and password. This information is passed back to the server, which in turn passes it to the security exit.

The security exit can be written to check many things. The sample user exit (**GTSECFTP**) checks RACF permissions. After processing, the security exit will notify the server whether or not it is OK to display the requested file.

The security exit is identified in the {Authorization} section of the **GTAIPFTP.INI** file.

Set Up the Security Exit

Simply put, to set up security you will:

1. Write the exit program to validate requests for files within the secured realm(s),

2. Update the **GTAIPFTP.INI** file to identify the name of the security exit program and the secured realm(s).

View the Sample Security Exit

The SAMPLIB contains a sample user exit (**GTSECFTP**) written in COBOL. You can use the comments in this sample program to help you understand the program requirements.

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Sample of {user} and {profile} entries in the **GTAIFTP.INI** file. Comments appear in blue.

```
{Users}
; Format is user=profile.
Brian=boss
Jeff=one
Janella=one
Bill=two
*=two
; The *=two entry covers any users
; who don't have a specific entry.
;
;Profiles
;
{one}
password=URM
; The security URM program is responsible
; for all security.
;
; No commands specified, default to 1
; (allow, no log).
;
{boss}
password=*
; No password required.
"STOR"=2
"RMD"=2
"RNFR"=2
"RNTO"=2
; The default command security is 1, which
; allows the command without logging. Users
; with the "boss" profile can use any command,
; but only the STOR, RMD, RNFR and RNTO
; commands are logged.
;
{two}
; No password value specified, entry defaults to "*".
ACCT=0
ALLO=-1
APPE=-1
DELE=-1
LIST=0
MKD=-1
MODE=0
RETR=2
RMD=-1
RNFR=-1
RNTO=-1
SMNT=-1
STAT=0
STOR=-1
STOU=-1
; All other commands have default value of 1
; (allow, no log).
```

Contents:

Related topics:

- BMS/TS Bridge Web Services
- BWS Server Administration
- FTP Server

Administrator's Guide Home
BMS/TS Doc. Home

Resolving Domain Names

This section explains how to set up BMS/TS Bridge Web Services (BWS) to resolve domain names.

BWS is able to use Internet addresses in dotted decimal format. However, in order to use names rather than the numeric addresses, BWS uses the RESOLVE.INI file.

Understanding the RESOLVE.INI File

Domain names are resolved using the entries in the RESOLVE.INI (initialization) file. This file is located in the **GT:/ETC** directory. All entries appear under the group heading **{DNS}**. The table below explains the parameters within the {DNS} group. The default value, if any, is shown in the parameter field.

Parameter/ Default Value	Description
buf=0	Size of the buffer for DNS queries, in bytes.
domain=	Domain name suffix, such as "gtsoftware.com". This is used to repeat a DNS look–up; if the first DNS search fails, the domain= DNS suffix will be appended to a "single word" request and the attempt will be repeated.
hosts=0	Use HOSTS.INI look–up? 0=No, 1=Yes.
id=0	Optional numeric value for DNS queries.
port=53	DNS port number.
server=	DNS address in dotted decimal format.
wait=4	Number of seconds to wait for response from DNS server.

Using the RESOLVE.INI File

Your use of the RESOLVE.INI file depends upon whether you have a Domain Name Server (DNS). The DNS associates a host name with its numeric address. Refer to the appropriate instructions below.

Using RESOLVE.INI with Your DNS

If you have a Domain Name Server, you must specify the **Server=** and **Domain=** parameters in the RESOLVE.INI file. The other parameters are optional.

Using RESOLVE.INI without a DNS

If you do not have a DNS, you must specify the **Domain=** parameter, and set the **Hosts=** parameter to '**1**'. This tells the browser to resolve names locally using the HOSTS.INI file.

Using the HOSTS.INI File

The HOSTS.INI file, located in the **BWS:/ETC** directory, associates a host name with its numeric value. This file is required only if you do not have a Domain Name Server.

All entries in the HOSTS.INI file appear in the {DNS} group. The format for entries in this file is ***host.name=IP_address***. For example:

www.gtsoftware.com=206.67.241.16

You will make an entry in HOSTS.INI for each host name you need to resolve. You can create and maintain the HOSTS.INI file using the command line processor Edit function or a text editor.

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Initialization (INI) Files

This document provides information about the BMS/TS initialization (INI) files, including:

- A table listing the INI files (in alphabetical order) which describes their purpose and location.
- A table listing references to the INI files in the BMS/TS documentation.
- Information about using and modifying the INI files.

BMS/TS INI Files

File Name	Description	Location
bms_bws.ini	Configuration file for parameters used by the BMS/TS Bridge Web Services	BWS:/ETC/
gtbipftp.ini	Configuration file for general purpose parameters used by the FTP server	BWS:/ETC/
gtbipsrv.ini	Configuration file for general purpose parameters used by the FTP listener	BWS:/ETC/
hfs.ini	Settings for the Hierarchical File System (HFS)	HFS:/
hosts.ini	Used with resolve.ini to do DNS name look–ups	BWS:/ETC/
protocols.ini	Sets protocol ID numbers	BWS:/ETC/
resolve.ini	Sets DNS look–up parameters	BWS:/ETC/
services.ini	Defines default port settings for various Internet services/protocols	BWS:/ETC/

The following table lists BMS/TS documents that refer to the INI files. This is not intended to be a complete list; refer to the appropriate index for additional references.

File Name	References
bms_bws.ini	BWS Server Administration Bridge Web Services
gtbipftp.ini	BWS Server Administration FTP Server Administration FTP Server Considerations
gtbipsrv.ini	BWS Server Administration FTP Server Administration FTP Server Considerations
hfs.ini	Hierarchical File System Cmd Line Processor: DIR Command

hosts.ini	Resolving Domain Names FTP Server Administration
protocols.ini	FTP Server Administration
resolve.ini	Resolving Domain Names FTP Server Administration
services.ini	FTP Server Administration

Understanding INI Files

The entries (variables) in initialization files are under specific sections which the server program checks during startup. Sections are indicated by headings within curly brackets, as follows: **{***name***}.**

Entries consist of value pairs (x=y) which are the code variables that control server operation. For example, if you put in a redirect for a URL shortcut for gtsoftware, the entry in the {Redir} section would look like this:

```
{Redir}
 /gtsoftware/*=http://www.gtsoftware.com
```

Modifying the Initialization Files Modifying initialization files should only be done by advanced users or under the direction of GT Software Technical Support. The entries are program variables, so they must be accurate and follow the exact syntax necessary for the program.

If it is necessary to edit an initialization file on a PC, use a text editor such as Notepad. Do not use a word processor because it may create errors. This section includes:

- Using the Server Initialization File Syntax,
- Entry Example,
- Initializing the File Settings.

Initialization File Syntax

When you edit an INI file, keep in mind these rules:

- Use curly braces "{ }" for section headings. Braces [ ] and **< >** are also recognized. Do not use parenthesis ( ).
- Use the existing section headings.
- Put entries under the proper headings.
- Put each entry on a new line. Don't string entries together using punctuation.

- Use an equal sign (**=)** between variables in value pairs. (Each line contains a value pair.)

- Use a semicolon (;)preceding each line of comments. (Do not use other code conventions, such as # or /* or // or < ! or –– .)

- Use the proper case for the user ID and password entries (they are case sensitive).

- Any entry that follows the *x=y* value pair pattern can use wildcards:
  - ♦ Use an asterisk (*****) to indicate 0 or more characters,
  - ♦ Use a question mark (?) to indicate 1 character.

See the example for an illustration of syntax and wildcards.

Note:

- Spaces are ignored.
- The initialization files are not case sensitive (except for user ID and password entries).
- See specific instructions at the beginning of each section in the initialization file.

Entry Example

Following is an example of an initialization file section illustrating properly made entries and the use of comments:

```
{DEFAULT_DOCS}
;assign either a filename extension to search for
;or * for an index
1=default.htm
2=index.htm
3=index.html
4=*
```

An entry `*.htm*` would result in any HTM or HTML file in the requested directory being returned as an index document.

Initializing New File Settings After changing values in an initialization file you need to restart (shutdown and then start) the server for the new settings to take effect.

To stop the FTP server, use the transaction GTRV SHUTDOWN.
To start the server, use the transaction GTRV.

The Bridge Web Services are automatically restarted when called by CICS.

Top of page

Administrator's Guide Home

Hierarchical File System (HFS)

The **hierarchical file system** (HFS) lets you use a VSAM file to organize your 3270 bridge–related files.

With HFS, you can define volumes, directories, and files in a VSAM file just as you might organize files on a PC hard disk or network.

HFS.INI File

The HFS.INI file is located in the **HFS:** volume. It contains settings that affect the hierarchical file (HFS) system. This file is supplied as part of the BMS/TS installation. You can customize the file settings as needed.

Entries in the INI files are grouped into related settings for easier use. Each group begins with a group name which appears in curly brackets ({*heading*}). The table below lists the valid group names for the INI files.

> **Note**: In this document, group headings are enclosed in "curly" brackets (French braces). In your HFS.INI file, you may need to use "square" brackets or "greater–than" and "less–than" signs, depending on your display type.

Group Settings Name	Description
{*volname*} **Note**: This may be any valid volume name, including the default volumes BWS: and HFS:.	Controls local cache and security settings for each volume.
{ALIAS}	Lets you create aliases for HFS commands.
{ASSIGN}	Assigns request for a volume name to another volume name. For example, you can "assign" the TEST: volume to the MYVOL: volume. If you request a list of directories in the TEST: volume, the system displays the directories in the MYVOL: volume, including those assigned from TEST:.

{DEFAULT}	Specifies defaults for common HFS settings.
{MOUNT}	Allows you to automatically mount the volumes on files other than the GT$FILE. **Note**: Volume names must be unique across the $files.
{REDIRECT}	Redirects access of a volume to a different *xx*$file. This allows you to use multiple VSAM files.
{ZIPTYPES}	Controls zip and unzip settings.

Rules for Editing the HFS.INI File

- You can use any text editor to view or modify the HFS.INI file.
- A semicolon in the first column indicates that the line is a comment.
- Lines with a left bracket in the first column contain the name of the group settings.
- All lines *other than comments and headings* must either be blank, or contain a file setting.
- File settings must begin in column 1.
- File settings must use the format *keyword*=*value*. Do **not** type a space on either side of the equal sign!

HFS Features

HFS offers a Command Line Processor much like a traditional PC command prompt. To complement that, HFS provides a powerful security system so you can easily set up CICS for your intranet, all the while maintaining as much security as you want over your intranet files.

Command Line Processor

The Command Line Processor provides a simple interface to HFS. From the command prompt, you can issue a few simple commands to define the entire structure of your CICS intranet.

Refer to Command Line Processor for instructions and an explanation of the available commands.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

HFS Security

The HFS internal security manager (ISM) provides widely–used "owner/group" and "permission" methods to protect your files. These methods are commonly used on a number of systems (such as UNIX) where intranet–related files are stored. The ISM is *always* available for use in restricting or enabling access to the files of your CICS intranet.

ISM provides security using an access control concept which associates users with groups. The access rights given to any user or group are detailed in a "permissions list". All HFS files, directories, and volumes have an Owner, a Group, and a Permissions List.

Owner, Group, Other

Each file, directory, and volume has a single *owner* –– the person who initially created the file, directory, or volume. And, as each user is associated with a group, each file, directory, and volume is also associated with the owner's *group*.

> Here's an example: suppose user **Bill** (who belongs to the group ***Users***) creates a file named INDEX.HTM. The file INDEX.HTM is owned by **Bill** and belongs to the group ***Users***.

Anyone other than the owner or the owner's group is referred to as "*other*".

Permissions List

The permissions list lets you enable or restrict access to a file, directory, or volume. The permissions list contains three sets of access flags. These flags grant access to **R**ead, **W**rite and e**X**ecute for the Owner, the Group and all Others.

Here's an example:

```
drw–rw–r–– BILL USERS 11/12/96 13:35:33 1 66116 policies
–rw–rw–r–– BILL USERS 11/13/96 09:43:55 1 139393 tech.html
–rw––––rwx BILL USERS 11/13/96 09:44:12 1 95474 activity.lst
```

The first field of each line contains the permissions list for that item. It's the line that looks like "drw–rw–r––".

The first character in this field is the resource type. Valid entries are "d" (directory), "–" (file), and "l" (locked file or directory). Notice in the example that "policies" is a directory.

The remaining nine characters in the field are viewed in groups of three. The next three characters are the Owner's permissions. The following three characters are the Group permissions; these are the rights given to anyone in the Owner's group. The last three characters are the permissions for all others; that is, anyone who is not the Owner and not in the Owner's group.

Each permission set is formatted the same way. An "r" in the first position indicates "read" access for the file/directory. A "w" in the second position indicates "write" access, which includes the ability to delete the file/directory. An "x" in the third position indicates "execute" access. A hyphen in any of these positions indicates that permission is not granted.

You can use the Command Line Processor to change file and directory attributes:

- Use the CHOWN command to change the owner of a file or directory.
- Use the CHGRP command to change the group of a file or directory.
- Use the CHATT command to change attributes (permissions list) of the file or directory.

You must have Write access to the resource to use the **CHOWN**, **CHGRP** or **CHATT** commands.

> Caution: Be careful using these commands. You can accidentally make a resource inaccessible to yourself!

Top of page

User's Guide Home
BMS/TS Doc. Home

Sample HFS.INI File

```
 {REDIRECT}
 ;map request for GT: to be on the GT$FILE dataset
 GT:=DDN:GT$FILE
 TED:=DDN:TEDFILE
 ;
 {ASSIGN}
 PRT:=GT:
 TED:=GT:
 ;
{ZIPTYPES}
 ; file pattern = {text | binary},{ascii | ebcdic},
 ; {compression level}
 ; example: *.bat=text,ASCII,4
 *Bat=text,ASCII,x Text file, convert to/from ASCII, use
default
 *.cgi=text,ASCII,x Text file, convert to/from ASCII, use
default
 *.cmd=text,ASCII,x Text file, convert to/from ASCII, use
default
 *.com=binary,,4 Binary file,,compression level 4
 *.exe=binary,,4 Binary file,,compression level 4
 *.gif=binary,,4 Binary file,,compression level 4
 *.htm*=text,ASCII,x Text file, convert to/from ASCII, use
default
 *.ini=text,ASCII,x Text file, convert to/from ASCII, use
default
 *.jp*g=binary,,0 Binary file,,compression level 0 (stored)
 *.sys=text,ASCII,x Text file, convert to/from ASCII, use
default
 *.zip=binary,,0 Binary file,,compression level 0 (stored)
 ;
 {GT:}
 CACHE=50
 READONLY=1
 NOACCESS=1
 ;
 {SYSTEM:}
 CACHE=50
 ;
 {TED:}
 CACHE=10
 ;
 {ALIAS}
 1=HELP          Display command help
 2=DIR           Directory display
 UNMNT=UNMOUNT   Alias for UNMOUNT command
 3=EDIT          Edit
 4=BROWSE        Browse
 5=FTP           File Transfer Process
 ;
```

```
{DEFAULT}
;0 = Short format
;1 = Brief format
;2 = Long format
DIR=2
;0 = MDY
;1 = DMY
;2 = YMD
DATE=0
DATESEP=-
TIMESEP=:
;
; Attributes for Owner, Group, World
OWNERATT=6
GROUPATT=6
WORLDATT=4
OWNER=GT
GROUP=PUBLIC

{MOUNT}
;mount volumes on files other than the GI$FILE
VOLUMES=2
;create work1: volume on AT$FILE
VOL1=AT$FILE\work1:
; create work2: volume on BT$FILE
VOL2=BT$FILE\work2:
```

Contents:

Related topics:

- Hierarchical File System

User's Guide Home
BMS/TS Doc. Home

Command Line Processor

The Command Line Processor allows you to manage files and perform administrative functions using a familiar PC–type command prompt.

Start the Command Line Processor

The transaction code to access the Command Line Processor is **GTC**. When you enter GTC from a blank CICS screen, the Command Line Processor screen displays (view sample).

To exit the Command Line Processor, press F3.

> *Hint:* The Command Line Processor buffer stores your command input and the resulting displays. You can easily review the information in the buffer by using the F7 and F8 keys to scroll through the buffer's contents. This allows you to see what you have done, or review the results of a previous operation. You can clear the screen buffer using the CLS command.

Command Summary

The table below summarizes the commands available in the Command Line Processor. Select the link for detailed information about each command.

HFS Command	Description
BROWSE	View the contents of a file. You cannot edit in browse mode.
CHATT	Change the attributes (permissions) of a file, directory or volume.
CHDIR	Change the working directory.
CHGRP	Change the name of the group to which the file, directory or volume belongs.
CHOWN	Change the name of the owner of the file, directory or volume.
CLS	Erase the display buffer and clear the screen.
CODEPAGE	Set the internal code page number associated with the pathname.
CONVERT	Convert the internal code page for the pathname.
COPY	Copy a file.

DELETE	Delete a file.
DIR	Display a list of the files (and subdirectories) within the specified directory or volume.
ECHO	Display messages, or toggle ECHO on/off.
EDIT	Execute external program to edit a file.
FTP	File Transfer Program, used to upload (import) files to the VSAM file, or download (export) files from the VSAM file.
HELP	Display online help for the command prompt; displays a list of valid commands and parameters.
LANG	Set the command shell output to display in an alternate language.
LISTZIP	Display the contents of a zipped file.
MKDEV	Create a new volume.
MKDIR	Create a new directory.
MOUNT	Associate a volume with an *xx*$FILE other than the default *xx*$FILE. This association remains until you issue the UNMOUNT command, or until the end of the current session.
PRINT	Print the source document to the designated printer.
RENAME	Rename a file within the same directory or to another directory.
RMDEV	Remove device (delete volume).
RMDIR	Remove (delete) directory.
TYPE	Display the contents of a file.
UNLOCK	Remove the lock placed on a file that was being updated at the time of an abend or other failure.
UNMOUNT	Remove the temporary association between a volume and an *xx*$FILE, as created by the MOUNT command.
UNZIP	Unzip (decompress) the contents of a zipped file.
VERSION	Display information about the software, including the current release level.
VOLS	Display listing of all volumes. *Hint:* To access a volume, type the volume name on the command line and press Enter. Be sure to type the colon (**:**) at the end of the volume name.
WHOAMI	Display the logged–on user's ID and group.
ZIP	Create a zipped (compressed) file.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: BROWSE

Use the BROWSE command to execute an external program to display the specified file.

BROWSE		
Command Format	**BROWSE *pathname***	
Parameters	*pathname*	The complete path to the file you wish to view, including volume, directory and filename.

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: CHATT

Use the CHATT command to change the attributes (permissions) of a file or directory.

CHATT (Change Attributes)	
Command Format	**CHATT *pathname nnn***
Parameters	*pathname* The complete path to the file, including volume, directory and filename. *nnn* The file permissions (attributes), expressed as a 3–digit number. The first digit identifies the permissions for the file's owner, the second digit is the permissions for the owner's group, and the third digit is the permissions for all others. To determine the numeric value of the permissions, combine the value for each of the rights allowed: Read access = 4 Write access = 2 Execute access = 1 For example, to grant the file owner Read/Write/Execute (4+2+1) access, the owner's group has Read/Execute (4+1) access, and others have Execute–only (1) access, the "*nnn*" portion of the CHATT command would be 751.

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: CHDIR

Use the CHDIR command to change the current working directory.

CHDIR (Change Directory)		
Command Format	**CHDIR *pathname***	
Parameters	*pathname*	The complete path to the desired directory, including the volume and higher–level directories.

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: CHGRP

Use the CHGRP command to change the name of the group to which the file/directory belongs.

CHGRP (Change Group)		
Command Format	**CHGRP *pathname group***	
Parameters	*pathname*	The complete path to the file/directory, including volume, directory and filename.
	group	The name of the new group.

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: CHOWN

Use the CHOWN command to change the name of the owner of the file/directory.

CHOWN (Change Owner)		
Command Format	**CHOWN *pathname owner***	
Parameters	*pathname*	The complete path to the file/directory, including volume, directory and filename.
	owner	The user ID of the new owner.

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: CLS

Use the CLS command to clear the screen and erase the display buffer.

CLS (Clear Screen)	
Command Format	**CLS**
Parameters	There are no parameters for this command.
Notes	The display buffer records user input and results. This information accumulates in the buffer until the CLS command is issued, at which time the buffer is cleared. *Hint:* How is this buffer useful? It allows you to review what you've done, or the results of a previous command. For example, if you issue a DIR (directory) command, often the results will not fit on a single screen. Because the information is retained in the buffer, you can use the F7 and F8 keys to scroll through the directory list, rather than re–issuing the command.

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: CODEPAGE

Use the CODEPAGE command to set the internal code page number associated with the pathname.

CODEPAGE		
Command Format	**CODEPAGE {–q|g} pathname codepage**	
Parameters	*–q*	(optional) Quiet. Do not display messages while copying.
	–g	(optional) Display the current code page settings for the specified files.
	pathname	The pathname may be a single filename (such as my.htm) or a path (such as GT:/WWW/*.HTM). The wildcard characters (* and ?) are accepted.
	codepage	The code page must be a valid code page number between 0 and 9999.

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: CONVERT

Use the CONVERT command to convert the internal code page to the specified code page.

CONVERT		
Command Format	**CONVERT {–q} pathname codepage**	
Parameters	*–q*	(optional) Quiet. Do not display messages while copying.
	pathname	The pathname may be a single filename (such as my.htm) or a path (such as GT:/WWW/*.HTM). The wildcard characters (* and ?) are accepted.
	codepage	The code page must be a valid code page number between 0 and 9999.

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: COPY

Use the COPY command to copy the specified file.

COPY		
Command Format	**COPY** *–q* ***frompath*** *topath*	
Parameters	*–q*	(optional) Quiet. Do not display messages while copying.
	frompath	The complete path to the file you wish to copy, including volume, directory and filename.
	topath	(optional) If you do not specify a new path (*topath*), the file is copied to the current directory.

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: DELETE

Use the DELETE command to delete the specified file.

Note: This command will not delete a directory; use the RMDIR command to remove a directory.

DELETE	
Command Format	**DELETE *path/filename***
Parameters	*path/filename* The complete path and filename of the file you wish to delete.
Notes	This command does not apply to directories or volumes. Use the RMDIR command to remove a directory; use the RMDEV command to delete a volume.

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: DIR

Use the DIR command to display the files and directories within the current volume/directory, or the volume/directory specified by the *pathname*.

DIR (Directory)	
Command Format	**DIR** –{b|c|s|l} *pathname*
Parameters	*–b* (optional) Brief list. Display file and directory names only. *–c* (optional) Use **–cc** to display code page numbers instead of block counts. Use **–cb** to display block counts. See note below for additional information. *–s* (optional) Display a short list. *–l* (optional) Display a long list, including permissions. *pathname* (optional) The complete path to the directory you wish to view, including volume and higher–level directories, if other than the current volume/directory.
Notes	The results of a DIR command will also display a count of the subdirectories and files listed. *The default value is set in the HFS.INI file as DIRCOUNT=. If DIRCOUNT=0, the system displays block counts. If DIRCOUNT=1, the default is code page number display. You can add DIRCOUNT to the {DEFAULT} section of the HFS.INI file.*

Sorting the Directory List	
By default, the results of the DIR command are sorted by name in ascending order (a through z). Subdirectories are displayed before file names. You can specify other sort criteria by using the following parameters:	
dir –on	Sort by name in ascending order. This is the default sort order.
dir –o–n	Sort by name in descending order.
dir –od	Sort by date in ascending order.

dir OD	Sort by date in descending order.
dir –os	Sort by size, smallest file first (subdirectories still sort by name).
dir Os	Sort by size, largest file first (subdirectories still sort by name).

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: ECHO

Use the ECHO command to display messages (*characters*), or toggle the ECHO on/off.

ECHO	
Command Format	**ECHO *characters***
Parameters	*characters* The message to display.

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: EDIT

Use the EDIT command to execute external programs that give you access to a text editor.

Note: You can use the editor to modify the file specified by the pathname.

EDIT		
Command Format	**EDIT *pathname***	
Parameters	*pathname*	The complete path to the file you wish to edit, including volume, directory and filename.

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: FTP

Use the FTP command to execute external file transfer programs to upload and download files.

FTP (File Transfer Program)	
Command Format	**FTP**
Parameters	There are no parameters for this command.
Notes	Some special characters, such as { and }, can become corrupted during a file transfer. One way to avoid this is to zip the file and transfer it as a binary (rather than a text) file.

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: HELP

Use the HELP command to display a list of all valid commands.

HELP	
Command Format	**HELP**
Parameters	There are no parameters for this command.

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: LANG

Use the LANG command to display command output in an alternate language (defined in NLS), or return display to default output (English).

LANG (Language)	
Command Format	**LANG *option***
Parameters	*option* One of the following: STD, 1, ALT, or 2. **LANG STD** (or LANG 1) is the default; all text displays in English. LANG ALT (or LANG 2) changes the display to the alternate text defined in NLS.

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: LISTZIP

Use the LISTZIP command to display the contents of a zipped file.

LISTZIP	
Command Format	**LISTZIP *zipfile***{–c|i|l|s|x|z}
Parameters	*zipfile* The path/filename of the zipped file. –c (optional) Display comments for the included files. –i (optional) Include the specified file(s). –l (optional) Long list. This is the default. –s (optional) Short list. –x (optional) Exclude the specified file(s). –z (optional) Display comments for the zipped file.

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: MKDEV

Use the MKDEV command to create a device (volume) and allocate storage.

MKDEV (Make Device)	
Command Format	**MKDEV *ddname volname: mbytes version***
Parameters	*ddname* The name of the *xx*$FILE associated with the volume. *volname:* The volume name. Be sure to include the colon (:) after the volume name! *Mbytes* The amount of storage allocated. *version* The version for the new volume. The default is **2**.

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: MKDIR

Use the MKDIR command to create a new directory.

MKDIR (Make Directory)	
Command Format	**MKDIR** ***pathname*** *owner group*
Parameters	*pathname* The complete path to the file you wish to view, including volume, directory and filename. *Owner* (optional) The owner of the new directory. *Group* (optional) The owner's group.

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: MOUNT

Use the MOUNT command to associate a volume with a *xx*$FILE other than the default.

MOUNT	
Command Format	**MOUNT *ddname volname:***
Parameters	*ddname* The *xx*$FILE you want to associate with the volume. *volname:* The name of the volume. **Note:** Be certain to include the colon (**:**) at the end of the volume name!

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: PRINT

Use the PRINT command to execute external programs to print a file.

PRINT		
Command Format	**PRINT** ***file*** *printer*	
Parameters	*file*	The complete path to the file you wish to view, including volume, directory and filename.
	printer	(optional) The logical printer you want to print to, if other than the default printer.

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: RENAME

Use the RENAME command to change the name of a file or directory.

You can also use this command to change the path only, keeping the current filename.

RENAME	
Command Format	**RENAME *frompath topath***
Parameters	*frompath* The complete path to the file you wish to view, including volume, directory and filename. *topath* The complete new path for the file/directory.

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: RMDEV

Use the RMDEV command to remove (delete) the specified device.

RMDEV (Remove Device)	
Command Format	**RMDEV *ddname volume:***
Parameters	*ddname* The name of the *xx*$FILE associated with the volume. *volume:* The volume name. Be sure to type the colon (**:**) at the end of the volume name!

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: RMDIR

Use the RMDIR command to remove (delete) the specified directory.

Note: You can only delete a directory if it is empty.

RMDIR (Remove Directory)	
Command Format	**RMDIR *pathname***
Parameters	*pathname* The complete path to the directory you wish to delete, including volume and higher–level directories.

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: TYPE

Use the TYPE command to display the contents of the specified file.

TYPE	
Command Format	**TYPE *pathname***
Parameters	*pathname* The complete path to the file you wish to view, including volume, directory and filename.

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: UNLOCK

Use the UNLOCK command to reset the file lock.

UNLOCK		
Command Format	**UNLOCK *pathname***	
Parameters	*pathname*	The complete path to the file you wish to unlock, including volume, directory and filename.

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: UNMOUNT

Use the UNMOUNT command to temporarily remove the association created between a volume and an *xx*$FILE with the MOUNT command.

UNMOUNT		
Command Format	**UNMOUNT *volname:***	
Parameters	*volname:*	The name of the volume to unmount. Caution: Be sure to type the colon (**:**) at the end of the volume name!

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: UNZIP

Use the UNZIP command to extract the contents of a zipped file. The zip file remains unchanged.

UNZIP	
Command Format	**UNZIP *zipfile*** {–b|c|d|l|o|q|s|t|v|z}
Parameters	*filename* The name of the file to unzip. If the file is not in the current directory, you must specify the complete path as well as the filename. –b (optional) Force binary. –c (optional) Display comments for the included files. –d (optional) Destination directory, if other than the current directory. –l (optional) Long list format. –o (optional) Overwrite existing files. –q (optional) Quiet. Do not display progress messages while unzipping. –s (optional) Short list. This is the default value. –t (optional) Force text. –v (optional) Verify the contents of the zipped file. –z (optional) Display the comments for the zipped file.

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: VERSION

Use the VERSION command to display the internal version.

VERSION	
Command Format	**VERSION**
Parameters	There are no parameters for this command.

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: VOLS

Use the VOLS command to display the volume list for the specified *xx*$FILE.

Hint: To access a specific volume, type the volume name on the command line and press Enter. Be sure to type the colon (**:**) at the end of the volume name.

VOLS (Volumes)		
Command Format	**VOLS** *–b ddname*	
Parameters	*–b*	(optional) The "brief" switch, which provides just a list of volume names.
	ddname	(optional) The *xx*$FILE where the volume resides.
Notes	The VOLS: command also displays the version number associated with each volume.	

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: WHOAMI

Use the WHOAMI command to display the user ID and the group of the current user.

WHOAMI	
Command Format	**WHOAMI**
Parameters	There are no parameters for this command.

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Command Line Processor

User's Guide Home
BMS/TS Doc. Home

Command Line Processor: ZIP

Use the ZIP command to create a zipped file. By default, the included files remain unchanged in their original directories.

ZIP		
Command Format	**ZIP *zipfile***{–b|d|f|i|m|p|q|r|t|x} ***filenames***	
Parameters	*zipfile*	The name of the zip file to be created. If you do not want the zipped file stored in the current directory, you must specify the complete path, as well as the file name.
	filename	The path/filename of the files to be included in the zipped file.
	–b	(optional) Force binary.
	–d	(optional) Delete the specified file(s) from the zip file.
	–f	(optional) Freshen. Overwrites existing files, but does not include new files in the zipped file.
	–i	(optional) Include the specified file(s) in the zipped file.
	–m	(optional) Move the specified file(s) into the zipped file, and delete the original file.
	–p	(optional) Store pathnames.
	–q	(optional) Quiet. Do not display progress messages during the zip process.
	–r	(optional) Recurse into directories. This allows you to include subdirectories in the zipped file. In most cases, you will also want to specify the –p parameter to maintain the pathnames.
	–t	(optional) Force text.
	–x	(optional) Exclude the specified file(s) from the zipped file.

Keep in mind:

- Required parameters appear in **boldface** print. Optional parameters appear in normal type, and are identified as optional in the parameter description.
- You must replace parameters displayed in *italics* with your own values.
- Commands are ***not*** case–sensitive.

Top of page

User's Guide Home
BMS/TS Doc. Home

Contents:

Password Administration

Password Administration is used to add and maintain the BMS/TS system password and feature codes. These passwords are required in order to use BMS/TS. The encrypted system password and feature code are provided by GT Software.

The transaction code to access the standalone Password Administration function is **GTIB**.

When you enter **GTIB** from a blank CICS screen, the Password Administration screen displays (view sample).

Password Administration Screen

The table below describes the fields on the Password Administration screen.

Field Name	Description
CPU Serial Number	The CPU ID, as used by GT Software to generate the SYSPASS and FEATURE codes. When you access the Password Administration screen, the CPU Serial Number field displays the serial number of the current CPU.
System Password	The 8–character system password (SYSPASS), provided by GT Software.
Feature Code	The 8–character feature code, provided by GT Software.
Checksum	Code used internally by BMS/TS.

The function keys listed below are available from the Password Administration screen.

Function Key	Description
F1 (Help)	Access the online help system.
F2 (Add)	Add a new CPU (and corresponding System Password and Feature Code).
F3 (Exit)	Exit the Password Administration function and return to a blank CICS screen.
F4 (Update)	Change the System Password, Feature Code, and/or Checksum for an existing CPU ID. **Note**: You cannot update a CPU ID.
F5 (Delete)	Delete an existing CPU ID.

	Note: You cannot delete the system password or feature code for an existing CPU ID; you must delete the entire CPU record.
F6 (Refresh)	Displays the values for the System Password, Feature and Checksum for the specified CPU ID.
F7 (Prev)	Displays the previous CPU ID record, if any.
F8 (Next)	Displays the next CPU ID record, if any.

Add a CPU ID

1. Type the serial number in the CPU Serial Number field.
   Remember to use the CPU ID as specified by GT Software.
2. Type the System Password.
3. Type the Feature Code.
4. Type the Checksum.
5. Press F2 to add the new CPU ID. You should receive the message Requested operation complete, and the password expiration date will display on the right side of the window.

To exit the Password Administration function, press F3.

Change the System Password and/or Feature Code

1. If you have multiple CPUs, enter the CPU ID and press F6 to refresh the screen.
2. Type the System Password (if changed).
3. Type the Feature Code (if changed).
4. Type the Checksum (if changed).
5. Press F4 to enter the changes. You should receive the message Requested operation complete.

Caution: You cannot change a CPU ID. You must add the new CPU and then delete the old CPU ID.

To exit the Password Administration function, press F3.

Delete a CPU ID

If you have defined only one CPU ID, you cannot delete it. You'll need to add the new CPU ID, and then you can delete the old CPU ID.

1. If the CPU displayed is not the CPU you want to delete, enter the CPU ID and press F6 to refresh the screen.
2. Press F5. The system prompts you to confirm the request to delete the CPU.

Caution: You cannot delete the System Password or Feature Code for an existing CPU ID.

To exit the Password Administration function, press F3.

Top of page

Printer Administration

The Printer Administration functions allows you to define and maintain printers.

The transaction code to access the standalone Printer Administration function is **GTP**.

When you enter GTP from a blank CICS screen, the Printer Administration Menu displays (view sample). To select an option from this menu, type the option number and press Enter.

Add a Printer Definition

Printer definitions are stored in the **PRT:** volume. The name format is ***prtname*.DEF**, where *prtname* is the logical printer name.

When you select the Add Printer Definitions option from the Printer Administration Menu, the Add Printer Definitions dialog displays. The table below describes the fields on this screen.

Field Name	Description
Logical Printer	The 1– to 8–character logical name for the printer. This is the name you will use in maintaining the printer definition.
Physical Printer	The CICS ID for the printer or queue.
Printer Type	Valid values are **P** (printer), **JES** (JES output queue), **TD** (Transfer Data), and **TS** (Temp Storage). The default value is **P**. If you specify **JES**, the system displays the JES Printer Definitions dialog. Contact your CICS Administrator if you need assistance.
Printer Length	The length (in rows) of the printer form. Valid values are 24 to 100. The default is **66**.
Printer Width	The width (in columns) of the printer form. Valid values are 80 to 255. The default is **80**.
Form Feed Before/After	These fields indicate whether you want to feed a blank form before and/or after the print job. Valid values are **Y** (yes) and **N** (no). The default is **N**.
Remote SYSID	Allows you to identify a printer in a different CICS region. By default, the Remote SYSID field is blank.
Route List	Indicates whether you want to route the print jobs from this printer to other printers/queues. The default value is **N** (no).

	If you specify yes, the system displays the Printer Route List Definitions dialog. This dialog allows you to specify up to 15 additional printers/queues.
3270 Data Streams	*This field is currently not used.*
Graphic Data Stream	*This field is currently not used.*

When you have finished defining a printer, press F3 to save the printer definition and exit to the Printer Administration menu. If you do not want to save the printer definition, press F12 to cancel.

To exit the Printer Administration menu, press F3.

Change a Printer Definition

When you select the Change Printer Definitions option from the Printer Administration Menu, the Change Printer Definitions dialog displays.

Enter the Logical Printer name and press Enter to display the Change Printer Definitions dialog. The fields on this dialog are the same as the fields on the Add Printer Definitions dialog.

> **Note**: If you do not know the logical name of the printer definition you want to change, press F2 to display the List Printers screen. You can select the desired printer from the list of all defined printers.

Modify the fields as needed, then press F3 to save the changes and exit to the Change Printer Definitions dialog. If you do not want to save the changes, press F12 to cancel.

To exit the Printer Administration menu, press F3.

Delete a Printer Definition

When you select the Delete Printer Definitions option from the Printer Administration Menu, the Delete Printer Definitions dialog displays.

Type the Logical Printer name and press Enter to delete the printer definition. The system displays the message Record Deleted, and clears the Logical Printer field. You can enter another printer name to delete, or press F3 to exit the Delete Printer Definitions dialog.

> **Note**: If you do not know the logical name of the printer definition you want to delete, exit this screen and select List Printers option. This allows you to view a list of all currently defined printers.

List Printers

When you select the List Printers option from the Printer Administration Menu, the system displays a list of all currently–defined printers.

From the list, you can select a printer definition to change. The selected printer definition displays on the Change Printer Definitions dialog.

To exit the List Printers screen without making a selection, press F3.

To exit the Printer Administration menu, press F3.

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Contents:

National Language Support (NLS) Administration

National Language Support (NLS) allows you to display Novation screens and messages in a language other than English.

> **Note:** BMS/TS is part of the Novation family of products, and it includes Novation functions such as HFS and online password administration. The screens for the Novation functions use NLS; these are the screens and messages you can modify. These Novation functions are also part of other GT Software products, such as ASSIST/TS.

The transaction code to access the standalone NLS Administration function is **GTLB**.

Understanding NLS Internal Tables

Enabling NLS involves translating the text of three internal tables. These tables contain the language–dependent text for Novation screens:

Table Name	Description
Command Table	Contains the valid commands that the user can enter on the command line of Novation screens.
Text Table	Contains the literals displayed on Novation dialogs and screens.
Message Table	Contains the error and informational messages issued by Novation.

The default language for these tables is **English**. To enable these tables for NLS, you must first translate them to the desired alternate language.

Temporary Storage Queue

The first time you use BMS/TS, the NLS internal tables are automatically loaded into CICS storage. At this time BMS/TS automatically stores a pointer to the NLS block in temporary storage queue. The format is **G$xJyyzz**, where:

x – is x'02' for BMS/TS,
yy – is the first two characters of the *xx*$FILE (usually "GT").
zz – is high values or x'FFFF'. Each GT$FILE supports English and one alternate language. If multiple

VSAM files are in use, then multiple NLS tables will also exist. Each NLS environment uses approximately 40K bytes of shared storage and a unique temporary storage queue name to hold a pointer to that environment.

The temporary storage queues used by NLS must be in the same CICS region from which NLS is loaded. Or, if a QOR (Queue–Owning Region) is in use, there must be only one AOR in which NLS can run. The temporary storage queue holds the pointer to the NLS tables loaded into memory in *each* AOR. If multiple AORs are executing NLS, the temporary storage queue might not contain the correct pointer for the current AOR.

Translating the NLS Tables

We recommend that you have someone who is familiar with the Novation functions do the editing/translating. To avoid errors, we recommend also that you first translate the tables on hardcopy before editing the tables online.

After you install BMS/TS, the GT$FILE contains the three internal English language tables (Command, Text, and Message). We refer to these tables as "internal" tables because they are never manually modified. When the same tables are translated to an alternate language, we refer to them as having been "externalized" because the tables *are* altered during the translation process.

Translation Procedure

There are three steps to translating the tables. After you have completed these steps, the BMS/TS interface will support both English and another language.

Step 1: *Copy the internal English tables to the external tables.*

Select the Merge Active Tables to Work Area for Editing option. This copies the current tables into the work area and allows to you edit the tables. **Step 2**: *Translate the external table entries to the alternate language.*

Translate the Command, Text, and Message tables. There is a separate menu option for each of these tasks: Edit the Command Table, Edit the Error Message Table and Edit the Text Table. **Step 3**: *Merge the external table entries back into the internal tables to create a new combined internal table.*

After you have edited/translated all three tables in the work area, you merge the work area tables with the active tables and then reload the active tables. The resulting three tables in the GT$FILE replace the original English–only tables.

Using the NLS Menu

When you enter GTLB from a blank CICS screen, the NLS Administrator's Menu displays (view sample). The table below describes the menu options you can select.

Menu Option	Description
Load default NLS tables	Resets the active tables to the default values.

	Note: Any changes you may have made to the tables will be lost.
Merge active tables to work area for editing	Creates a duplicate of the active tables in the work area. This allows you to work on the tables without effecting the currently active tables.
Merge work tables with active tables	Merges the updated tables from the work area with the active tables. **Note**: You need to reload the tables before the changes take effect.
Edit command table in work area	Accesses the Command Table Editor.
Edit error message table in work area	Accesses the Message Table Editor.
Edit text table in work area	Accesses the Text Table Editor.
Reload active tables	Reloads the active tables to VSAM, and then into memory. **Note**: Changes you have made in the work area and merged into the active tables will not take effect until you reload the active tables.

Load Default Tables

This option allows you to reset the active tables to the default values.

> **Note**: Any changes you may have made to the tables will be lost.

When you select the Load Default Tables option, BMS/TS loads the default tables and displays the message Tables have been reloaded.

You can make another selection from the NLS menu, or press F3 to exit.

Merge Active Tables to Work Area for Editing

This option allows you to create a duplicate of the active tables in the work area. You can then work on the tables without effecting the currently active tables.

When you select the Merge Active Tables to Work Area for Editing option, BMS/TS copies the active tables to the work area, and then displays the message Operation Complete.

You can make another selection from the NLS menu, or press F3 to exit.

Merge Work Tables with Active Tables

This option allows you to merge the updated tables from the work area with the active tables.

When you select the Merge Work Tables with Active Tables option, BMS/TS merges your edited tables from the work area into the active tables, and then displays the message Operation Complete.

You can make another selection from the NLS menu, or press F3 to exit.

Remember: You must reload the tables before the changes take effect.

Edit the Command Table

The Edit Command Table option allows you to edit/translate the commands on Novation screens.

When you select the Edit Command Table option from the NLS Menu, the Command Table Editor screen displays (view sample). This table contains the Novation commands, in English.

The table below describes the fields on the Command Table Editor screen.

Field Name	Description
Num	Indicates the internal command number. You cannot edit this field.
LN	Indicates the total character length of the translated command. For example, if you translated the English command ALL to ALLES, you would place a 5 in this field because ALLES contains 5 characters.
AB	Indicates the minimum number of characters that must be entered for the command in the **Command** field. Set this field to the same value as the data in the **LN** field to force the user to enter the entire command word. Or, to allow abbreviations of the command word, enter the desired abbreviation length for the command word. For example to allow users to abbreviate the command ALLES to ALL, you would enter a 3 in this field.
Command	Indicates the desired translated command. For example, you might want to translate the English command ALL to ALLES.

1. If desired, place the cursor in the Desc field and add your own description for the translated Command Table Editor screen. You can enter up to 35 characters.

2. Use the blank line beneath each entry to translate the command text. Use the F7 and F8 keys to scroll forward and backward through all commands. If at any time during the translation process you wish to cancel your changes and return to CICS, press F12 or the Clear key.

3. Press F4 to merge the external Command Table entries back into the internal tables. The Table

has been merged message displays when the process is complete.

4. Press F3 to return to the blank CICS screen.

Edit the Error Message Table

The Edit Error Message Table option allows you to edit/translate the text of Novation error messages.

When you select the Edit Error Message Table option from the NLS menu, the Message Table Editor screen displays (view sample). This screen displays BMS/TS system messages, in English.

The table below describes the fields on the Message Table Editor screen.

Field Name	Description
Num	Indicates the internal message number. You cannot edit this field.
Standard Text / Alternate Text (if any)	Indicates the desired translated message text. For example, you might want to translate the English message "This book does not have an Index" to "Dieses Buch hat keinen Index".

1. If desired, place the cursor in the Desc field and add your own description for the translated Message Table Editor screen. You can enter up to 35 characters.

2. Use the blank line beneath each entry to translate the message text. Use the F7 and F8 keys to scroll forward and backward through all messages. If at any time during the translation process you wish to cancel your changes and return to CICS, press F12 or the Clear key.

3. Press F4 to merge the external Message Table entries back into the internal tables. The Table has been merged message displays when the process is complete.

4. Press F3 to return to the blank CICS screen.

Edit the Text Table

The Edit Table Text option allows you to edit/translate the literals that are used to write BMS/TS screens and dialogs.

When you select the Edit Text Table option from the NLS menu, the Text Table Editor screen displays (view sample). This screen displays the literal characters used to write out BMS/TS dialogs and screens.

The table below describes the fields on the Text Table Editor screen.

Field Name	Description
Num	Indicates the internal text number. You cannot edit this field.
Standard Text / **Alternate Text** (if any)	Indicates the desired translated literal text. For example, you might want to translate the English literal "2. No" to "2. Nein." **Note**: To avoid potential display problems, whenever possible use the same number of characters in your translated text as appears in the English text. If necessary, you can use embedded blanks to accomplish this.

1. If desired, place the cursor in the Desc field and add your own description for the translated Text Table Editor screen. You can enter up to 35 characters.

2. Use the blank line beneath each entry to translate the literal text. Use the F7 and F8 keys to scroll forward and backward through all literals. If at any time during the translation process you wish to cancel your changes and return to CICS, press F12 or the Clear key.

3. Press F4 to merge the external Text Table entries back into the internal tables. The Table has been merged message displays when the process is complete.

4. Press F3 to return to the blank CICS screen.

Reload Active Tables

When you select the Reload Active Tables option, BMS/TS reloads the active tables to VSAM, and then into memory.

> **Note**: Changes you have made in the work area and merged into the active tables will not take effect until you reload the active tables.

After reloading the tables, BMS/TS displays the message Operation Complete. You can select another option from the NLS menu, or press F3 to exit.

Top of page

Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Supported Code Pages

This document lists the code pages currently supported by BMS/TS.

EBCDIC Code Pages	
Code Page	**Country**
37	US and English–speaking Canada
273	Austria, Germany
277	Denmark, Norway
278	Finland, Sweden
280	Italy
284	Latin America, Spain
285	United Kingdom
297	France
500	International
875	Greek
924	Multinational with euro support (variation of code page 500 and 1047)
1140	US/Canada/Netherlands with euro support
1141	Austria/Germany with euro support
1142	Denmark/Norway with euro support
1143	Finland/Sweden with euro support
1144	Italy with euro support
1145	Spain with euro support
1146	United Kingdom with euro support
1147	France with euro support

ASCII Code Pages	
Code Page	**Country**
437	US English

813	Greek ISO 8859–7
819	Western ISO 8859–1
850	ASCII multilingual
858	IBM multinational with euro (variation of code page 850)
912	Central European ISO 8859–2
923	ISO 8859–15 (variation of ISO 8859–1 with euro and Icelandic support)

Top of page

Administrator's Guide Home
ASSIST/TS Doc. Home

Contents:

Appendix A, Using the Line Editor

A line editor enables you to create, review, and update files containing lines of text. These textual lines perform a variety of functions: they set parameters, define operating procedures, protect data with passwords, etc. With a line editor such as the one included with BMS/TS, you work with one line of text at a time.

BMS/TS provides a line editor so you can perform a number of tasks. Although the body of the editor may vary depending on the option you are performing, the format of the different screens is basically the same.

This document describes how to perform options using the editor's command line, line commands, and function keys.

Line Editor Functions

You will use the BMS/TS line editor to:

- Edit input rules,
- Create and update tables,
- Edit the BMS/TS Transaction Monitor,
- Edit JCL,
- Update Operator Security.

The BMS/TS line editor provides three ways to process information on the editor screen:

- Line Command Fields: Enter commands which insert, copy, move, and/or delete lines of text from the editor.
- Command Prompt: Perform specific operations by entering commands from a Command line.
- Function Keys: Perform specific operations by pressing a function key.

Both the Command Prompt and the Function Keys enable you to locate and change data, display help, move around the editor screen, save information, and exit the editor screen. (View sample editor screen.)

With the exception of the JCL Editor, each of the editor screens has input fields for specific entries relating to the option. The JCL Editor has a blank body that allows you to write and review JCL

members.

Entering a Line Command

You can enter a line command to insert, copy, move, and/or delete lines of text from the editor.

1. Press Tab to move to the desired line in the LC column.
2. Perform the desired function:

If you want to...	Then...
Insert a line	1. Type I (Insert). 2. Press Enter. BMS/TS inserts a blank line immediately after the line in which the cursor is positioned.
Insert multiple lines	1. Type I*nn* (where *nn* indicates the number of lines you want to insert, from 1–99). 2. Press Enter. BMS/TS inserts the desired number of blank lines immediately after the line in which the cursor is positioned.
Copy a line	1. Type R (Replicate) or " (double quotation mark). 2. Press Enter. BMS/TS copies the line on which the cursor is positioned.
Copy multiple lines	1. Type R*nn* or *nn* (where *nn* indicates the number of lines you want to copy, from 1–99). 2. Press Enter. BMS/TS copies the desired number of lines beginning from the position of the cursor.
Copy a block of lines	1. Type CC in front of the first line you want to copy. 2. Type CC again in front of the last line in the block you want to copy. 3. Position the cursor where you want to copy the marked block of lines. 4. Type A (After) or B (Before) to indicate whether you want to insert the block after or before the current line. 5. Press Enter. BMS/TS copies the block of lines to the desired location.
Move a line	1. Type M (Move) in front of the line you want to move. 2. Position the cursor where you want to move the line. 3. Type A (After) or B (Before) to indicate whether you want to insert the line after or before the current line. 4. Press Enter. BMS/TS moves the line to the desired location.
Move a block of lines	1. Type MM (Move) in front of the first line you want to move. 2. Type MM again in front of the last line in the block you want to move.

	3. Position the cursor where you want to copy the marked block of lines. 4. Type A (After) or B (Before) to indicate whether you want to insert the block after or before the current line. 5. Press Enter. BMS/TS moves the block of lines to the desired location.
Move current line to top of screen	1. Type / (forward slash). BMS/TS redisplays the screen with current line at the top of the editor screen. Keep in mind that this is to help text entry only; the screen arrangement actually remains unchanged.
Delete a line	1. Type D (Delete). 2. Press Enter. BMS/TS deletes the line where your cursor is positioned.
Delete multiple lines	1. Type D*nn* (where *nn* indicates the number of lines you want to delete, from 1–99). 2. Press Enter. BMS/TS deletes the desired number of lines beginning from where your cursor is positioned.

Using the Command Prompt

You can perform specific functions (such as repositioning the cursor, changing data, etc.) by entering processing commands from the Command line.

1. Press Tab to move to the Command line at the bottom of the screen.
2. Perform the desired function:

If you want to...	Then...
Move to the bottom of the editor screen	1. Type BOT or B (Bottom). 2. Press Enter. BMS/TS moves the cursor to the last line of the screen.
Move to the top of the editor screen	1. Type TOP or T (Top). 2. Press Enter. BMS/TS moves the cursor to the first line of the screen.
Move down the editor screen a certain number of lines	1. Type D*nn* (where *nn* indicates the number of lines you want to move down, from 1–99). 2. Press Enter. BMS/TS moves the cursor down the desired number of lines.
Move up the editor screen a certain number of lines	1. Type U*nn* (where *nn* indicates the number of lines you want to move up, from 1–99). 2. Press Enter. BMS/TS moves the cursor up the desired number of lines.
Scroll the display forward	1. Type SF (Scroll Forward). 2. Press Enter. BMS/TS moves the display forward one screen (if applicable).

Scroll the display backward	1. Type SB (Scroll Backward). 2. Press Enter. BMS/TS moves the display back one screen (if applicable).
Switch between half and full screen scrolling	1. Type SC (Scroll). 2. Press Enter. BMS/TS switches the type of screen scrolling (from half page to full page, or vice versa).
Switch between regular and alternate display	1. Type FS (Format Switch). 2. Press Enter. BMS/TS switches the type of screen display (from regular to alternate display, or vice versa, if applicable). **Note:** Alternate display lets you review information such as the current operator ID, length of a physical map, etc.
Display the Help information for the screen	1. Type HELP or H (Help). 2. Press Enter. BMS/TS displays help information for the appropriate editor screen.
Locate a specific line number	1. Type L*nn* (where *nn* indicates the number of the line you want to locate, from 1–99). 2. Press Enter. BMS/TS moves the cursor to the desired line.
Locate a specific string of data	1. Type L /*data* or L '*data*' (where *data* represents the data you want to search for in the editor screen). 2. Press Enter. BMS/TS searches for and moves the cursor to the first occurrence of the desired data. **Note:** BMS/TS searches forward from the position of the cursor.
Repeat the last locate data command	1. Type RL (Repeat Locate). 2. Press Enter. BMS/TS searches for and moves the cursor to the next occurrence of the desired data. **Note:** BMS/TS searches forward from the position of the cursor.
Change data	1. Type CHG /*olddata*/*newdata*/ or CHG '*olddata*'*newdata*' (where *olddata* represents the existing string of data that you want to change, and *newdata* represents the new string of data to use to replace the old data). 2. Press Enter. BMS/TS searches for the first occurrence of the *olddata*, and replaces it with the *newdata*.
Repeat change of data	1. Type RC (Repeat Change). 2. Press Enter. BMS/TS repeats the last change data command.
Change all occurrences of data	1. Type CHG /*olddata*/*newdata*/**/ or CHG '*olddata*'*newdata*'**' (where *olddata* represents the existing string of data that you want to change, and *newdata* represents the new string of data to use to replace the old

	data). 2. Press Enter. BMS/TS searches for all occurrences of the *olddata*, and replaces them with the *newdata*. **Note:** To ensure that you change all desired data, move the cursor to the top of the editor screen before using this command.
Sort transaction codes or table entries in ascending order	1. Type SORT. 2. Press Enter. BMS/TS sorts the transaction codes on Transaction Monitor Control screen, or table entries, in ascending order.
Cancel the edit and return to the previous menu	1. Type CANCEL or CAN (Cancel). 2. Press Enter. BMS/TS returns to the previous menu without saving the current changes to the data on the editor screen.
Save data on screen and return to previous menu	1. Type END or E (End), or FILE or F (File). 2. Press Enter. BMS/TS saves the updates to the editor screen and returns to the previous menu.

Using Function Keys

You can perform specific functions (such as displaying help information, repositioning the cursor, etc.) by pressing a function key. The following table describes the available function keys. Note that some function keys are not available on all screens.

If you want to...	Then...
Display the Help information for the screen	Press F1 or F13. BMS/TS displays help information for the appropriate editor screen.
Switch between regular and alternate display	Press F2 or F14. BMS/TS switches the type of screen display (from regular to alternate display, or vice versa, if applicable). **Note:** Alternate display lets you review information such as the current operator ID, length of a physical map, etc.
Scroll right to display all 80 columns of Job Control Language	Press F2 or F14. BMS/TS scrolls right so you can see all 80 columns of the screen. **Note:** This only applies to the JCL Editor.
Save data on screen and return to previous menu, or exit help screen	Type F3 or F15. BMS/TS saves the updates to the editor screen and returns to the previous menu, or exits the help screen and returns to the editor screen.
Repeat the last locate data command	Press F5 or F17. BMS/TS repeats the last locate data command.

	Note: This key option does not appear on the Transaction Monitor Control screen.
Repeat the last change of data command	Press F6 or F18. BMS/TS repeats the last change data command. **Note:** This key option does not appear on the Transaction Monitor Control screen.
Scroll the display backward	Press F7 or F19. BMS/TS moves the display back one screen (if applicable).
Scroll the display forward	Press F8 or F20. BMS/TS moves the display forward one screen (if applicable).
Switch between half and full screen scrolling	Press F9 or F21. BMS/TS switches the type of screen scrolling (from half page to full page, or vice versa).
Move cursor to the command line	Press F12 or F24. BMS/TS moves the cursor to the Command line at the bottom of the editor screen.
Process data keyed on screen and/or move to next screen	Press Enter. BMS/TS processes the data you entered on the editor screen and either lets you key more data or continues to the next screen.
Cancel the edit and return to previous menu	Press Clear. BMS/TS returns to the previous menu without saving the current changes on the editor screen.

Top of page

BMS/TS Doc. Home

Contents:

Appendix B, Batch Functions

This appendix provides basic information about the BMS/TS batch functions. It describes how to use the batch functions, with examples of JCL showing how to run maintenance jobs, conversions, and how to generate BMS macro source code.

This appendix provides instructions for using the following BMS/TS batch programs:

- GTBCDMS –– import/convert a DMS panel,
- GTBCONV –– import a map to BMS/TS,
- GTBPUNCH –– generate BMS source code ,
- GTBUTIL –– perform batch file utility operations.

Contents:

Related topics:

- Batch Functions

User's Guide Home
Administrator's Guide Home
BMS/TS Doc. Home

Using GTBCDMS to Convert a DMS Panel

You can bring a DMS panel into the BMS/TS system library so that you can perform maintenance on the map and use it with standard CICS programs. You convert DMS panels using the GTBCDMS panel conversion program.

BMS/TS converts your DMS panels according to the following routine.

1. GTBCDMS reads the working panel the DMS card images as input.
2. GTBCDMS builds the symbolic mapname from the type "E" record of the DMS panel.
3. GTBCDMS builds a BMS/TS map of the panel. The conversion function also produces an audit trail to help you locate problems in a conversion.

   Note: Keep in mind that DMS allows some parameters and conventions such as duplicate field names that are invalid for regular programming languages. Any errors associated with these invalid entries appear on the audit trail.
4. After BMS/TS converts the panel, the PRINT batch function lets you print an image of the map with the position of every field, field name, and attributes.

Convert DMS Panels

You can convert your DMS panel into a BMS/TS map so that you can perform maintenance on the map.

1. Identify the DMS panels you want to convert.
2. Log–on to your editor.
3. Write JCL to convert your DMS panel into your BMS/TS system library so that you can perform maintenance on the map.

   Note: In order for the GTBCDMS program to work correctly, your JCL must include a DLBL or DD statement pointing to GT$FILE. Refer to the JCL samples below.

4. Run the job.

5. Review the audit trail to ensure that the panel successfully converted.

6. Review the converted map online to verify that you can access and update it with BMS/TS. See Working With Maps for information on how to review and/or update a map.

Sample JCL for GTBCDMS

Sample JCL (OS/390) for GTBCDMS

```
//BMSTS    JOB. . . .
//JOBCAT   DD   DSN=usercat,DISP=SHR
//GTBCDMS  EXEC PGM=GTBCDMS
//STEPLIB  DD   DSN=BMSTS.LOADLIB,DISP=SHR
//SYSLST   DD   SYSOUT=A
//GT$FILE  DD   DSN=GT.MASTER.FILE,DISP=SHR
//SYSIN    DD   *
 DMS001                                      C  004A
                                             2  A
1210                                         1  02B
TEST DMS PANEL                               2  B
1          DMS PANEL dms001
2LINE2 : ____________FIELD1======>
                                                C
                                                D
FLD1    21010    FLD2    23515A                 E
/*
/&
```

Sample JCL (VSE) for GTBCDMS

```
// JOB. . . .
// DLBL GT$FILE,'GT.MASTER.FILE',,VSAM
// EXEC PROC=GTBLIBS
// EXEC GTBCDMS,SIZE=AUTO
 DMS001                                      C  004A
                                             2  A
1210                                         1  02B
TEST DMS PANEL                               2  B
1               DMS PANEL dms001
2LINE2 : ____________FIELD1======>
                                                C
                                                D
FLD1    21010    FLD2    23515A                 E
/*
/&
```

Top of page

User's Guide Home
Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Batch Functions
- Appendix D, Support for Multi–Byte Character Sets

User's Guide Home
Administrator's Guide Home
BMS/TS Doc. Home

Using GTBCONV to Convert/Import BMS Maps

You can import a BMS map to the BMS/TS system library using OS/390 and VSE operating systems. Then you can review the audit trail and the converted map so that you can perform maintenance on the map.

You convert/import BMS maps using the GTBCONV map conversion program. The parameters used with GTBCONV vary depending on your operating system.

The conversion program will convert most BMS macro–produced maps and copybooks. SDF or maps produced by other screen generators may require minor manual edits.

> **Note:** Through the rest of this document, we will use the term “import” in place of “import/convert”.
>
> Caution: You must close the GT$FILE in order to use GTBCONV.

GTBCONV Import Process

BMS/TS imports your BMS maps according to the following routine.

1. GTBCONV first retrieves the desired physical map from the designated load library, then assigns "dummy" field names to the map. Because the conversion function does not require any BMS source macros, you can bring any map used by CICS into the BMS/TS system library.

2. GTBCONV extracts the symbolic map from the specified object or copybook library.

   Note: If the symbolic map is an input/output map, the mapname must include an I (Input) or O (Output) suffix. All fields within the symbolic map must also have the appropriate suffixes in order for BMS/TS to correctly identify field types during the conversion.

3. GTBCONV compares the field specifications of both the physical and symbolic maps to ensure that they correctly match.

4. If the field names match, GTBCONV places the field names from the symbolic map into the physical map. The resulting map is a BMS/TS map.

Importing BMS Maps – OS/390

Unlike the other BMS/TS batch functions, the parameters associated with GTBCONV do not immediately follow the GTBCONV command in the actual JCL. Instead, they are passed at appropriate points in the job stream so that BMS/TS can locate the desired physical and symbolic BMS maps that you want to import.

The following table describes the valid parameters when using GTBCONV with an OS/390 system.

Parameters	Description
DSN	**Required** –– Indicates the VSAM library where your BMS map is cataloged. Use the format: DSN=*xxxxxxxx* where *xxxxxxxx* is the library containing the BMS map.
DSN	**Required** –– Indicates the name of the library containing the physical BMS map you want to import. Use the format DSN=*yyyyyyyy* where *yyyyyyyy* is the library containing the physical map.
DCB	**Required** –– Defines the imported BMS map as having the same block size as the original. This avoids errors during the conversion. Use the format: DCB=BLKSIZE=*xxxx* where *xxxx* is the map block size.
CONVERT	**Required** –– Indicates 1) name of the map you want to import, and 2) the programming language in which map is written and field prefix of map if applicable. Use the format CONVERT MAPSET=*aaaaaaaa*,LANG=*b*,PREFIX=*c* where: • *a* is the 1–8 character map name • *b* is the name of the programming language you are using. Valid entries are: C (COBOL), A (Assembler), P (PL/1), and R (RPG). • *c* is the field prefix, if applicable **Note:** If your map includes a field prefix, you must use the CONVERT statement and the PREFIX parameter. By doing this, BMS/TS includes the prefix on the symbolic map, which you can then alter when you update the map with BMS/TS.
DSN	**Required** –– Indicates the name of the library containing the symbolic BMS map you want to import. Use the format DSN=*zzzzzzzz* where *zzzzzzzz* is the library containing the symbolic map.

Procedure –– Import BMS Maps (OS/390)

1. Identify the BMS maps you want to import.
2. Log–on to your editor.
3. Write JCL to import your BMS map into the BMS/TS system library so that you can perform maintenance on the map with your OS/390 system. Refer to the JCL sample below.

   Note: In order for the GTBCONV program to work correctly, your JCL must include a DD statement pointing to GT$FILE.
4. Run the job.
5. Review the audit trail to ensure that the map was successfully imported.
6. Review the imported map online to verify that you can access and update it with BMS/TS.

This JCL is only an example; your JCL must be specific to the files and libraries used at your company.

```
//.....   JOB. . . .
//GTBCONV EXEC PGM=GTBCONV
//STEPLIB DD   DSN=BMSTS.LOADLIB,DISP=SHR
//        DD   DSN=loadlib.of.executable.map,DISP=SHR
//SYSLST  DD   SYSOUT=A
//GT$FILE DD   DSN=GT.MASTER.FILE,DISP=SHR
//SYSIN   DD   *
 CONVERT MAPSET=CUSTACC,LANG=C,PREFIX=ACCT-
/*
//        DD   DSN=copybook.pds(copybook),DISP=SHR
```

Importing BMS Maps – VSE

Unlike the other BMS/TS batch functions, the parameters associated with GTBCONV do not immediately follow the GTBCONV command in the actual JCL. Instead, they are passed at appropriate points in the job stream so that BMS/TS can locate the desired physical and symbolic BMS maps that you want to import.

The following table describes the valid parameters when using GTBCONV with a VSE system.

Parameter	Description
FROM	**Required** –– Indicates the name of the library containing the symbolic BMS map you want to import. Use the format FROM=*xxxxxxxx* where *xxxxxxxx* is the library containing the symbolic map.
PUNCH	

	Required –– Indicates the programming language in which the BMS map is written, and the name of the BMS map you want to import. Use the format PUNCH *x.yyyyyyyy* where: • *x* – The name of the programming language you are using. Valid entries are: C (COBOL), A (Assembler), P (PL/1), R (RPG). • *y* – The 1–8 character map name.
SEARCH	**Required** –– Indicates the name of the library or libraries containing the physical BMS map you want to import. Use the format SEARCH=*xxxxxxxx,yyyyyyyy* where: • *xxxxxxxx* – the first library you want BMS/TS to search for the physical map, • *yyyyyyyy* – the second library you want BMS/TS to search for the physical map. **Note:** BMS/TS will search through all indicated libraries for the physical map, so include all desired libraries in the SEARCH parameter. When including multiple libraries, be sure to separate each library with a comma.

Procedure –– Import BMS Maps (VSE)

1. Identify the BMS maps you want to import.
2. Log–on to your editor.
3. Write JCL to import your BMS map into your BMS/TS system library so that you can perform maintenance on the map with your VSE system. Refer to the sample JCL below.

   Note: In order for GTBCONV to work correctly, your JCL must include a DLBL statement pointing to GT$FILE.
4. Run the job.
5. Review the audit trail to ensure that the map was successfully imported.
6. Review the imported map online to verify that you can access and update it with BMS/TS.

This is only an example; your JCL must be specific to the files and libraries used at your company.

```
// JOB. . . .
// DLBL IJSYSPH,'PUNCH.FILE',69/001
// EXTENT SYSPCH,DOSRES,. . . .
 ASSGN SYSPCH,DISK,. . . .
// LIBDEF SL,FROM=ATSDEMO
// EXEC SSERV
 PUNCH C.CUSTACC
/*
 CLOSE SYSPCH,PUNCH
```

```
// DLBL IJSYSIN,'PUNCH.FILE',69/001
// EXTENT SYSIPT,DOSRES,. . . .
 ASSGN SYSIPT,DISK,. . . .
// DLBL GT$FILE,'GT.MASTER.FILE',,VSAM,CAT=CUST
// EXEC PROC=GTBLIBS
// LIBDEF CL,SEARCH=ATSDEMO,ATXDEMO
*
// EXEC GTBCONV,SIZE=AUTO
 CLOSE SYSIPT,SYSRDR
/&
```

Importing BMS Maps – VSE/SP

Unlike the other BMS/TS batch functions, the parameters associated with GTBCONV do not immediately follow the GTBCONV command in the actual JCL. Instead, they are passed at appropriate points in the job stream so that BMS/TS can locate the desired physical and symbolic BMS maps that you want to import.

The following table describes the valid parameters when using GTBCONV with a VSE/SP system.

Parameter	Description
ACCESS	**Required** –– Indicates the name of the library containing the symbolic BMS map you want to import. Use the format: ACCESS S=*xxxxxxxx*, where *xxxxxxxx* is the library containing the symbolic map.
FORMAT	**Required** –– Indicates that the format of the PUNCH parameter places the programming language indicator last. Always **FORMAT=OLD**.
PUNCH	**Required** –– Indicates the name of the BMS map you want to import, and the programming language in which the BMS map is written. Use the format: PUNCH *yyyyyyyy.x*, where: • *y* – the 1–8 character map name, • *x* – the name of the programming language you are using. Valid entries are: C (COBOL), A (Assembler), P (PL/1), and R (RPG).

Procedure – Import BMS Maps (VSE/SP)

1. Identify the BMS maps you want to import.

2. Log–on to your editor.

3. Write JCL to import your BMS map to the BMS/TS system library so that you can perform maintenance on the map with your VSE/SP system. Refer to the sample JCL below.

   Note: In order for the GTBCONV program to work correctly, your JCL must include a DLBL statement pointing to GT$FILE.

4. Run the job.

5. Review the audit trail to ensure that the map was successfully imported.

6. Review the imported map online to verify that you can access and update it with BMS/TS.

This JCL is only an example; your JCL must be specific to the files and libraries used at your company.

```
// JOB. . . .
// DLBL IJSYSPH,'PUNCH.FILE',69/001
// EXTENT SYSPCH,DOSRES,. . . .
 ASSGN SYSPCH,DISK,. . . .
// EXEC SSERV
 ACCESS S=ATSDEMO
 PUNCH CUSTACC.C,FORMAT=OLD
/*
 CLOSE SYSPCH,PUNCH
// DLBL IJSYSIN,'PUNCH.FILE',69/001
// EXTENT SYSIPT,DOSRES,. . . .
 ASSGN SYSIPT,DISK,. . . .
// DLBL GT$FILE,'GT.MASTER.FILE',,VSAM,CAT=CUST
// EXEC PROC=GTBLIBS
// EXEC GTBCONV,SIZE=AUTO
 CLOSE SYSIPT,SYSRDR
/&
```

Review the Audit Trail

You can review the audit trail to verify that the BMS maps were successfully imported.

The BMS/TS map conversion utility produces an audit trail that you can use verify that the BMS maps were successfully imported, and to quickly locate problems. A sample audit trail appears below:

```
mmddyyyy          BMS/TS CONVERSION

      CONVERT MAPSET=COB3,LANG=C,PREFIX=ACCT-
     B538 NEW MAPSET NAME ACCEPTED
            01 C0B3I
     B546 AUTOMATIC RETRIEVE OF MAPSET : COB3 COB3

     B539 MAP FOUND
            02 ACCT-FLDTEST1I
     B541 EXISTING FIELD NAME : FLD001
     B535 NEW FIELD NAME ACCEPTED      : FLDTEST1

            02 ACCT-FLDTEST2I.
     B587 GROUP FIELD FOUND

     B541 EXISTING FIELD NAME          : FLD002
     B535 NEW FIELD NAME ACCEPTED      : FLDTEST2

            03 ACCT-GRPFLD3I

     B541 EXISTING FIELD NAME          : FLD002
     B535 NEW FIELD NAME ACCEPTED      : GRPFLD1

            03 ACCT-GRPFLD2I
```

```
    B541 EXISTING FIELD NAME          : FLD002
    B535 NEW FIELD NAME ACCEPTED      : GRPFLD2

           03 ACCT-FLDOOCCSI
    B586 FIELD OCCURS FOUND

    B541 EXISTING FIELD NAME          : FLD003
    B535 NEW FIELD NAME ACCEPTED      : FLDOCCS

B543 MAP REWRITTEN            : COB3 COB3
B510 ** OPERATION COMPLETE ** VM: 6.3.0
```

Audit Trail Error Messages

If the map import process does not successfully complete, the audit trail will include one or more error messages. These error messages identify the problem encountered during the conversion process.

The following table describes some of the error messages you are most likely to receive. (Refer to System Messages for a complete list of BMS/TS messages.)

Error Message	Description	Action
B530 Too many fields found	Occurs if the symbolic and physical maps do not match. Either: • the symbolic map has more fields than the physical map, or • the physical map has more fields than the symbolic map, or • you edited a physical map containing arrays but did not update the symbolic map.	Verify that the fields of the symbolic map and physical map match, then retry the conversion.
B544 Field size does not match	Occurs if the symbolic and physical maps do not match. Either: • you edited the size of a field for one map but did not update the other, or • a map contains a group field and the size of the members does not add up to the total number of bytes in the group.	Verify that the symbolic map is current and that it matches the physical map, then retry the conversion.
$302 Record not found	Occurs if BMS/TS cannot locate the physical map in the specified library.	Verify that the specified library contains the physical map, then retry the conversion.

Hint: If a mapset contains one map that is causing conversion errors, you can delete that map from the copybook and run the conversion again. Then you will need to manually convert only the map that caused the error.

Arrays and Audit Trail Error Messages

If you encounter an error on the audit trail that is related to an array problem, you must perform the following steps.

1. Remove the partially–imported mapset from GT$FILE. This is necessary because BMS/TS begins creating the imported file even if you subsequently encounter errors. The imported file therefore may not be complete.

2. Check your JCL, then retry the import program.

3. If the problem persists, update the partially–imported file. This enables you to correctly define all field, line, column, or user–defined arrays on the map using BMS/TS. Refer to Working With Maps for more information.

4. Retry the import program.

Review the Imported Map

Review the imported map to verify that all fields were defined correctly. Update the map as needed. Refer to Working with Maps for instructions.

> **Note:** Maps that contain occurs will return a condition code of 4; be sure to review these maps.

Top of page

User's Guide Home
Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Batch Functions

User's Guide Home
Administrator's Guide Home
BMS/TS Doc. Home

Using GTBPUNCH to Generate BMS Source Code

You generate BMS source code by using the GTBPUNCH batch macro program and the PUNCH function.

PUNCH Format

Use the following format for the PUNCH function:

PUNCH MAPSET=*aaaaaaa*

Note: Type a comma to separate each parameter from any following parameter.

Also, in order for the PUNCH function to work correctly, your JCL must include a DLBL or DD statement pointing to GT$FILE.

PUNCH Parameters

The following table describes the PUNCH parameters.

PUNCH Parameters	Description
MAPSET	**Required** –– Indicates the name of the mapset for which you want to generate BMS source code. **Note:** The mapsets for which you want to generate macro source code must already reside in the BMS/TS system library.
The following parameters are optional.	
BASE	Indicates whether you want to specify a base name for your COBOL or PL/1 programs. This allows symbolic maps of more than one mapset to use the same storage base.

	Use the format BASE=*aaaaaaaa* where *aaaaaaaa* represents the desired 1–8 character base name.
CNTRL	Indicates whether you want BMS/TS to generate the CATAL and BKEND for VSE or IEBUPDTE for OS/390 control cards. Valid entries are YES or NO. The default is **YES**.
LANG	Indicates the name of the programming language you are using. Valid entries are: C (COBOL), A (Assembler), P (PL/1), and R (RPG). The default is **C** (COBOL).
LDC	Indicates whether you want to use the CICS mnemonic when determining the logical device code to be used for BMS. Use the format LDC=*aa* where *AA* represents the CICS mnemonic.
MODE	Indicates whether you want to create a symbolic map for input or output use. Valid entries are: IN (create symbolic map for input use only), OUT (create symbolic map for output use only), and INOUT (create symbolic map for input & output). The default is **INOUT**.
OBFMT	Indicates whether you want to enable the BMS OBFMT option. Valid entries are YES or NO. The default is **NO**. **Note:** The BMS OBFMT option indicates that all maps within this mapset are eligible for use in outboard formatting.
PRINT	Indicates whether you want to enable the BMS print option for the map. Valid entries are YES or NO. The default is **NO**. Keep in mind that you must set this parameter to YES if you want to use the printer with this map.
SHORTNAME	Indicates that you want to use the short field name convention in the BMS source code. Include the SHORTNAME parameter to enable the short–naming convention. Otherwise, the long–naming convention is the default.
STORAGE	Indicates whether you want to set up each map within its own storage area. Valid entries are AUTO (Yes) or NO. The default is **NO**.
SUFFIX	Indicates whether BMS/TS should attach a single alphabetic character to the field name of the symbolic map. This allows you to use a suffix to describe the field. Valid entries are: • I –– Input • O –– Output • L –– Length • A –– Attribute • F –– Flag byte, if the field has been modified **Note:** BMS conventions require a suffix on field names. BMS/TS field name lengths are therefore short by one byte from standard field name lengths to accommodate the BMS requirement. In most cases, you should use one of the above suffixes so that the field name on your symbolic map

	conforms to CICS conventions.
TIOAPFX	Indicates whether you want to insert a terminal input/output area prefix for each map. Valid entries are YES or NO. The default is **YES**. **Note:** This parameter is required for command–level programming languages.
TYPE	Indicates whether you want to generate physical and/or symbolic maps during the assembly process. Valid entries are: • MAP (generate physical map only), • DSECT (generate symbolic map only), • &SYSPARM (generate physical & symbolic maps). The default is **&SYSPARM**.
ZEROLEN	Indicates whether you want to enable the BMS/TS zero field length capability. This allows you to set ending field attributes to a length of either 1 or 0. Valid entries are: YES (length of zero 0 for ending field attributes), and NO (length of one 1 for ending field attributes). The default is **YES** (zero).

PUNCH Examples

The following sample PUNCH function generates BMS macro source code for all members of the GTBDEMO mapset.

```
PUNCH MAPSET=GTBDEMO
```

The following sample PUNCH function generates BMS macro source code for all members of the GTBDEMO mapset. Notice that it generates code using the long naming convention but without using the BMS/TS zero length capability.

```
PUNCH MAPSET=GTBDEMO,LONGNAME,ZEROLEN=NO
```

PUNCH JCL Samples

The following examples show JCL statements that use the PUNCH function. Keep in mind that these are only examples; your JCL must be specific to the files and libraries used at your company.

Sample JCL (OS/390) for GTBPUNCH

```
//BMSTS    JOB. . . .
//JOBCAT   DD   DSN=usercat,DISP=SHR
//GTBPUNCH EXEC PGM=GTBPUNCH
//STEPLIB  DD   DSN=BMSTS.LOADLIB,DISP=SHR
//SYSLST   DD   SYSOUT=A
//GT$FILE  DD   DSN=GT.MASTER.FILE,DISP=SHR
//GT$PCHS  DD   DSN=&&PCHS,UNIT=SYSDA,SPACE=CYL,1,1,
                DISP=,PASS,DCB=BLKSIZE=80
```

```
//SYSIN    DD   *
PUNCH MAPSET=GTBDEMO
/*
//UPDATE   EXEC PGM=IEBUPDTE
//SYSPRINT DD   SYSOUT=A
//SYSUT1   DD   DSN=users.objlib,DISP=SHR
//SYSUT2   DD   DSN=users.objlib,DISP=SHR
//SYSIN    DD   DSN=*.GTBPUNCH.GT$PCHS,
//              DISP=OLD,DELETE,DELETE
/*
```

Sample JCL (VSE) for GTBPUNCH

```
// JOB. . . .
// DLBL GT$FILE,'GT.MASTER.FILE',,VSAM
// DLBL IJSYSPH,'PUNCH.FILE'
// EXTENT. . . .
 ASSGN SYSPCH,DISK,. . . .
// EXEC PROC=GTBLIBS
// EXEC GTBPUNCH,SIZE=AUTO
 PUNCH MAPSET=GTBDEMO
/*
 CLOSE SYSPCH,PUNCH
// DLBL IJSYSIN,'PUNCH.FILE'
// EXTENT. . . .
 ASSGN SYSIPT,DISK,. . . .
* ASSGN A PVT SOURCE LIBRARY IF NEEDED
// EXEC MAINT
 CLOSE SYSIPT,SYSRDR
/&
```

Sample JCL (VSE/SP) for GTBPUNCH

```
// JOB. . . .
// EXEC PROC=GTBLIBS
// DLBL GT$FILE,'GT.MASTER.FILE',,VSAM
// DLBL IJSYSPH,'PUNCH.FILE'.0
// EXTENT SYSPCH,. . . .
 ASSGN SYSPCH,DISK,. . . .
// EXEC GTBPUNCH,SIZE=AUTO
 PUNCH MAPSET=GTBDEMO
/*
 CLOSE SYSPCH,PUNCH
// DLBL IJSYSIN,'PUNCH.FILE'.0
// EXTENT SYSIPT. . . .
 ASSGN SYSIPT,DISK,. . . .
// EXEC LIBR,PARM='A S=user.source'
 CLOSE SYSIPT,SYSRDR
/&
```

Using PUNCH to Pass Information to the Output File

If desired, you can use PUNCH to pass information to the output file. This lets you add special comments, date cards, etc., to the BMS source code.

To pass a card to the VSE punch file or OS/390 GT$PCHS file, type a % (percent sign) in the first position. This shifts the card image one position to the left and writes to the output file. For example, the following sample JCL adds two comments indicated by asterisks * to the source code:

```
%* ASSEMBLY DATE=@DATE
% PUNCH MAPSET=GTBDEMO
%* MAPS FOR GENERAL DEMO
```

Top of page

User's Guide Home
Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- Batch Functions

User's Guide Home
Administrator's Guide Home
BMS/TS Doc. Home

Using the GTBUTIL Batch Utility Program

The GTBUTIL utility program is used to perform the batch operations described below.

GTBUTIL Batch Functions

You use the GTBUTIL utility program to perform the following batch operations.

Function	Description
ALTER *	Changes the options of the existing *ADMIN record; directs processing to alternate CICS libraries. (See also CREATE, MOVE and SETFILE.)
BACKUP **	Backs up the entire BMS/TS system library. (See also DISKSEL.)
CATAL	Moves a specific map into the BMS/TS system library. (See also PROD and PUNCH.)
COPY *	Makes a duplicate of a member in the current BMS/TS system library. (See also RENAME.)
CREATE *	Initializes the BMS/TS system library and creates the *ADMIN record. (See also ALTER.)
DELETE *	Removes a member from the current BMS/TS system library.
DISKSEL **	Selectively backs up the BMS/TS system library. (See also BACKUP.)
DLIST	Prints a list of the entire BMS/TS system library. (See also PRINT.)
MOVE *	Transfers a member from one CICS library to another. (See also ALTER and SETFILE.)
PASSWRD *	Adds/changes password of a member in the current system library.
PRINT	Selectively prints a list of the BMS/TS system library. (See also DLIST.)
PROD *	Changes the status of a member in a library from test to production. (See also CATAL and PUNCH.)
PUNCH	Moves a specific table into the BMS/TS system library. (See also CATAL and PROD.)
RENAME *	Changes name of a member in the current BMS/TS system library. (See also COPY.)
RESET *	

	Changes the update flag of a set and/or member in the current BMS/TS system library so that you can edit the set/member.
RESTORE *	Restores the previous BMS/TS system library backup. (See also TAPESEL.)
SETFILE	Redirects all input and output requests to a specific CICS library. (See also ALTER and MOVE.)
TAPESEL *	Selectively restores the previous BMS/TS system library backup. (See also RESTORE.)
*You must close the *xx*$FILE in order to use this command. **You must close the *xx*$FILE to updates to ensure that you get an accurate backup copy of the file.	

Top of page

User's Guide Home
Administrator's Guide Home
BMS/TS Doc. Home

GTBUTIL ALTER

Use the ALTER function to change the parameters of the existing *ADMIN record.

> Caution: You must close the GT$FILE to use the ALTER command.

You can modify some of these parameters using the online *ADMIN functions.

The ALTER function enables you to direct processing to CICS libraries other than GT$FILE.

ALTER Format

Use the following format for the ALTER function.

> ALTER *parameter(s)*

Note: Type a comma to separate each parameter from any following parameter.

Caution: If the *ADMIN record is protected by a master password (MastPass), then you must supply the MastPass every time you run the ALTER command.

In order for the ALTER function to work correctly, your JCL must include a DLBL or DD statement pointing to GT$FILE.

ALTER Parameters

The following table describes the ALTER parameters. All of these parameters are optional.

ALTER Parameter	Description
ALIGNED	Indicates whether the maps produced by BMS/TS should be aligned or unaligned. Valid entries are YES (aligned) or NO (unaligned). The default is **NO**.
CSSF	Identifies the transaction used for your sign–off support. Use this parameter if you need BMS/TS to interface with your SIGNON transaction to perform storage cleanup and termination after a forced log–off. The default is **NONE** (no sign–off cleanup). **Note:** If you use this parameter, you must change the PCT entry for your sign–off transaction to point to GTBCSSN.

CSSFP	Identifies your CICS sign−off program name. The default is **DFHSNP**. Use this parameter to tell BMS/TS if you are using a sign−off program name other than a standard IBM sign−off program. Without this information GT Software cannot service your CICS region after a CICS ABEND or a forced log−off.
CSSN	Identifies the transaction used for your sign−on support. The default is **NONE** (no signon cleanup). Use this parameter if you need BMS/TS to interface with your SIGNON transaction to perform storage clean−up and termination after a forced log−off.
CSSNP	Identifies your CICS sign−on program name. Use this parameter to tell BMS/TS if you are using a sign−on program name different from the standard IBM sign−on program. Without this information GT Software cannot service your CICS region after a CICS ABEND or a forced log−off. The default is **DFHSNP**.
DEMOLIB	Indicates the 2−letter prefix of the *DEMO library, which contains your demo members. There is no default.
DKEY	Indicates the function key F1 through F24 that you want to act as the secondary help key for field level help requests. This option affects all entries on the Transaction Monitor and is system wide. There is no default.
EUROPE	Indicates whether to enable the European options. These options include date as well as numeric editing. Valid entries are Y (Yes) or N (No). The default is **N**.
HELPMODEL	Indicates the format of all BMS/TS help transactions for the TOR Terminal Owning Region and AOR Application Owning Region. The format of this parameter is **X**H**, where: • **X** = any alpha or numeric character, • ***** = wildcard: any character is valid, • **H** = Always H. For example: G**H −− Use all BMS/TS transaction codes starting with G and ending with H in the TOR/AOR help process. GT*H −− Use all BMS/TS transaction codes starting with GT and ending with H in the TOR/AOR help process.
HELPTRN	Indicates the help transaction code if you plan to run BMS/TS in a MRO multi−region option environment and have BMS/TS installed in the AOR only. Use a 3−character transaction code. The default is **GTB**.
INTLIB	Indicates the internal library. The default is **GT**, which represents the DDNAME that points to the internal library containing the internal library members BMS/TS help screens and menus. **Note:** When you use a prefix other than GT, you must change the internal library prefix to reflect the new DDNAME prefix.

ITERM	Indicates the number of terminal work areas BMS/TS should acquire at system initialization. The value for this parameter should be an average for the number of active terminals using BMS/TS at the same time. There is no default.
JCLLIB	Indicates the 2–letter prefix of the library that contains your JCL members. The default is **GT**.
MAPLIB	Indicates the 2–letter prefix of the *MAPS library, which contains your maps. The default is **GT**.
MASTPASS	Allows you to enter the 1–8 character master password that protects various system maintenance functions. This also overrides any member passwords. **Note:** Do not include this parameter if you do not have a current master password.
MONITOR	Indicates the transaction code of the product that controls the Transaction Monitor. The default is **GTB**.
MRO	Required only if you are running BMS/TS in a MRO environment. Valid entries are: • **NONE** – You are running BMS/TS in a standalone environment. This is the default value. • **AOR** – You are running BMS/TS in an AOR environment; BMS/TS is not initialized in the TOR. You must also specify the parameter HELPMODEL=NONE. • **TORAOR** – BMS/TS will be initialized in both the TOR and AORs. You must also specify the HELPMODEL parameter. See the HELPMODEL parameter in this table for more information.
NAME	Indicates the name you want to assign to your library. The name can include up to 30 characters, but must begin and end with the same special character which cannot be used in the name itself. For example, if you wanted to call the library the Production Library, you would enter: +PRODUCTION LIBRARY+
NEWMAST	Allows you to replace the existing master password. **Note:** Use the word **NONE** to completely remove password security from your library.
PKEY	Indicates the function key F1 through F24 that tells BMS/TS to interpret the next function key pressed as a help request. This expands the number of function keys available for help. There is no default.
PRILIB	Indicates the primary library. The default is **GT**, which represents the DDNAME that is built based on the first two characters of the CICS transaction code for online functions.
SIGNON	Indicates whether to enable the security function. Valid entries are Y (Yes) or N (No). The default is **N**.
TCTUA	Required only if you are running BMS/TS in a MRO environment and installing BMS/TS in the TOR.

	The format is *nnn*, where *nnn* represents an offset into the TCTUA of a 8 byte area. BMS/TS uses the area specified to pass information between CICS regions. The default is **NONE**.

ALTER Examples

The following sample ALTER function defines a master password (MASTPASS) to the *ADMIN record.

```
ALTER MASTPASS=password
```

The following sample ALTER function defines the library name.

```
ALTER NAME=+production library+
```

The following sample ALTER function enables security for the TR$FILE library.

```
ALTER SIGNON=y,PRILIB=tr
```

Top of page

User's Guide Home
Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

GTBUTIL BACKUP

Use the BACKUP function to make a backup copy of the entire BMS/TS system library.

> Caution: You must close the *xx*$FILE to updates in order to ensure that you get an accurate backup copy of the file.

BACKUP Format

Use the following format for the BACKUP function.

```
BACKUP LIST=xxx
```

Note: Type a comma to separate each parameter from any following parameter.

In order for the BACKUP function to work correctly, your JCL must include a DLBL or DD statement pointing to GT$FILE.

BACKUP Parameters

The following table describes the BACKUP parameters. All of these parameters are optional.

BACKUP Parameter	Description
FROMDATE	Backup members created on/after the specified date. Enter the date in *ccyymmdd* format; this format does not change for European dates.
LIST	Produces a list of the members written to the backup tape. Valid entries are YES or NO.
REWIND	Indicates whether you want the tape rewound following the backup. Valid entries are YES or NO. The default is **NO**.
TODATE	Backup members created on/before the specified date. Enter the date in *ccyymmdd* format; this format does not change for European dates.

BACKUP Example

The following sample BACKUP function backs up the BMS/TS system library and instructs BMS/TS to include a list of members written to the tape.

```
BACKUP LIST=YES
```

BACKUP JCL Samples

The following examples show JCL statements that use the BACKUP function. Keep in mind that these are only examples; your JCL must be specific to the files and libraries used at your company.

BACKUP: Sample JCL (OS/390)

```
//BMSTS    JOB. . . .
//JOBCAT   DD   DSN=usercat,DISP=SHR
//GTBUTIL  EXEC PGM=GTBUTIL
//STEPLIB  DD   DSN=BMSTS.LOADLIB,DISP=SHR
//SYSLST   DD   SYSOUT=A
//GT$FILE  DD   DSN=GT.MASTER.FILE,DISP=SHR
//TAPOUT   DD   DSN=xxxxxx, UNIT=TAPE,. . . . .
//SYSIN    DD   *
 BACKUP
/*
```

Note: OS/390 can also back up to disk.

BACKUP: Sample JCL (VSE)

```
// JOB. . . .
// DLBL GT$FILE,'GT.MASTER.FILE',,VSAM
// TLBL TAPOUT,. . . . .
// ASSGN SYS008,cuu
// EXEC PROC=GTBLIBS
// EXEC GTBUTIL,SIZE=AUTO
 BACKUP
/*
/&
```

Top of page

User's Guide Home
Administrator's Guide Home
BMS/TS Doc. Home

GTBUTIL CATAL

Use the CATAL function to build a symbolic and/or physical map and catalog it into the BMS/TS system library.

> **Note:** Output of CATAL is two files unless otherwise stated in the function: one for the physical map, one for the symbolic map.

CATAL Format

Use the following format for the CATAL function.

CATAL MAPSET=*aaaaaaa*

Note: Type a comma to separate each parameter from any following parameter.

In order for the CATAL function to work correctly, your JCL must include a DLBL or DD statement pointing to GT$FILE.

CATAL Parameters

The following table describes the CATAL parameters.

CATAL Parameter	Description
MAPSET	Indicates the name of the mapset that you want to move to the BMS/TS system library. This is the **only required** CATAL parameter.
The parameters listed below are optional.	
ADS	Indicates whether you want to create ADS for the 3270 bridge. Valid entries are Y (Yes, create ADS for 3270 bridge) or N (No, do not create ADS). The default value is **N**.
BASE	Indicates whether you want to specify a base name for your COBOL or PL/1 programs. The format is *aaaaaaaa*, where *aaaaaaaa* represents the desired base name.

BRCOMP	Indicates whether you want to compress the HTML code. Valid values are Y (Yes, compress the HTML) or N (No, do not compress). The default value is **Y**. **Note:** HTML is generated, like the BMS macros, with each tag on its own line. This creates requirements for large buffer sizes used by the CICS Bridge. To reduce these buffer requirements, use BRCOMP=Y to combine tags on lines where room allows.
BRIDGE	Currently the only valid value is bridge=html (or bridge=h). This forces the ADS parameter to Y.
BRLOOK	Indicates how you want to the HTML screens to display. • BRLOOK=IBM produces a light gray background similar to that of the BMS macro template generation. • BRLOOK=GT produces a black background similar to a 3270−style display. This is the default value.
BRTYPE	Indicates the type of field names. • BRTYPE=1 represents simple field names as implemented in Transaction Server version 1. Type 1 example: &FLDNAME (uses full field name). • BRTYPE=2 represents the field naming conventions used in Transaction Server version 2 in *rrcccllll* (row, column, length) format. Type 2 example: &F010020032_FLDNAME (uses up to 21 characters of name). If you are unsure which to use, BRTYPE=1 is the only option supported by all versions of Transaction Server. Default is **1**.
CNTRL	Indicates whether you want BMS/TS to generate the CATAL and BKEND for VSE or IEBUPDTE for OS/390 control cards. Valid entries are YES or NO. The default is **YES**.
DONLY	Indicates whether you want to suppress the building of the physical map. Valid entries are YES or NO. The default is **NO**.
LANG	Indicates the name of the programming language you are using. Valid entries are: • C (COBOL) −− this is the default. • A (Assembler) • P (PL/1) • R (RPG) **Note:** You must set this parameter to “A” (Assembler) if you are requesting a resident map.
LDC	Indicates whether you want BMS/TS to use the CICS mnemonic when determining the logical device code to be used for BMS. Use the format LDC=*aa*, where *aa* represents the CICS mnemonic.
MAP	Indicates the name of the map that you want to move to the BMS/TS system library.

	Note: This parameter is only valid for Assembler programs.
MAPONLY	Indicates whether you want to suppress the building of the symbolic map. Valid entries are YES or NO. The default is **NO**.
MODE	Indicates whether you want to create a symbolic map for input or output use. Valid entries are: • IN –– Create symbolic map for input use only. • OUT –– Create symbolic map for output use only. • INOUT –– Create symbolic map for input & output use; this is the default.
OBFMT	Indicates whether you want to enable the BMS OBFMT option. Valid entries are YES or NO. The default is **NO**. **Note:** The BMS OBFMT option indicates that all maps within this mapset are eligible for use in outboard formatting.
PRINT	Indicates whether you want to enable the BMS print option for the map. Valid entries are YES or NO. The default is **NO**. You must set this parameter to YES if you want to use the printer with this map.
RES	Indicates whether the physical map is resident in the ALC program. Valid entries are YES or NO. The default is **NO**. If you set this parameter to YES, you must also use the MAP parameter. **Note:** This parameter is only valid for Assembler programs.
SOSI	Indicates whether you want all of the maps in the mapset to be MBCS/SOSI–enabled. Valid entries are Y (Yes, make all maps in mapset MBCS/SOSI–enabled) or N (No, do not make the maps in the mapset MBCS/SOSI–enabled). The default value is **N**.
STORAGE	Indicates whether you want to set up each map within its own storage area. Valid entries are AUTO (Yes) or NO. The default is **NO**.
SUFFIX	Specifies a single alphabetic character for BMS/TS to attach to the end of the name of the physical and symbolic maps.
SUPP01	Indicates whether you want to change the 01 levels in the symbolic map to 02 to accommodate the new compiler restrictions for COBOL copybooks. Valid entries are YES or NO. The default is **NO**.
TIOAPFX	Indicates whether BMS/TS should insert a terminal input/output area prefix for each map. Valid entries are YES or NO. The default is **YES**. **Note:** This parameter is required for command–level programming languages.

Mapset Restrictions

The IBM 3270 Bridge requires a suffix which is appended to the mapset name. If your mapset name is 7 characters, the suffix is limited to one additional character (A–Z, 0–9). This means that you can have no more than 36 maps in the mapset.

If your mapset name is 6 (or fewer) characters, the suffix can be two additional characters (A–Z, 0–9, AA–ZZ, 00–99). You can have as many as 72 maps in the mapset.

CATAL Example

The following sample CATAL function submits all members of the GTBDEMO mapset to the BMS/TS system library.

```
CATAL MAPSET=GTBDEMO
```

CATAL JCL Samples

The following figures show examples of JCL statements that use the CATAL function. Keep in mind that these are only examples; your JCL must be specific to the files and libraries used at your company.

Note: The SAMPLIB member CATAL includes samples showing the Bridge Web Services parameters.

CATAL: Sample JCL (OS/390)

```
//BMSTS    JOB. . . .
//JOBCAT   DD   DSN=usercat,DISP=SHR
//GTBUTIL  EXEC PGM=GTBUTIL
//STEPLIB  DD   DSN=BMSTS.LOADLIB,DISP=SHR
//SYSLST   DD   SYSOUT=A
//GT$FILE  DD   DSN=GT.MASTER.FILE,DISP=SHR
//GT$PCHP  DD   DSN=&&PCHP,UNIT=SYSDA,SPACE=CYL,1,1,
                DISP=,PASS,DCB=BLKSIZE=80
//GT$PCHS  DD   DSN=&&PCHS,UNIT=SYSDA,SPACE=CYL,1,1,
                DISP=,PASS,DCB=BLKSIZE=80
//SYSIN    DD   *
CATAL MAPSET=GTBDEMO
/*
//UPDATE   EXEC PGM=IEBUPDTE
//SYSPRINT DD   SYSOUT=A
//SYSUT1   DD   DSN=users.copylib,DISP=SHR
//SYSUT2   DD   DSN=users.copylib,DISP=SHR
//SYSIN    DD   DSN=*.GTBUTIL.GT$PCHS,
//              DISP=OLD,DELETE,DELETE
//LINK     EXEC PGM=IEWL,PARM='LIST,NCAL,LET, XREF'
//SYSPRINT DD   SYSOUT=A
//SYSLMOD  DD   DSN=BMSTS.LOADLIB,DISP=SHR
//SYSUT1   DD   UNIT=SYSDA,SPACE 1024, 100,10
//SYSLIN   DD   DSN=*.GTBUTIL.GT$PCHP,
//              DISP=OLD,DELETE,DELETE
/*
```

CATAL: Sample JCL (VSE)

```
// JOB. . . .
// DLBL GT$FILE,'GT.MASTER.FILE',,VSAM
// DLBL IJSYSPH,'PUNCH.FILE'
// EXTENT. . . .
 ASSGN SYSPCH,DISK,. . . .
// EXEC PROC=GTBLIBS
// EXEC GTBUTIL,SIZE=AUTO
 CATAL MAPSET=GTBDEMO
/*
 CLOSE SYSPCH,PUNCH
// DLBL IJSYSIN,'PUNCH.FILE'
// EXTENT. . . .
 ASSGN SYSIPT,DISK,. . . .
// EXEC PROC=GTBLIBS
* CATALOG PHYSICAL MAP
// OPTION CATAL
 INCLUDE
// EXEC LNKEDT
* CATALOG SYMBOLIC MAP
// EXEC MAINT
 CLOSE SYSIPT,SYSRDR
/&
```

CATAL: Sample JCL (VSE/SP)

```
// JOB. . . .
// DLBL GT$FILE,'GT.MASTER.FILE',,VSAM
// DLBL IJSYSPH,'PUNCH.FILE'
// EXTENT. . . .
 ASSGN SYSPCH,DISK,. . . .
// EXEC PROC=GTBLIBS
// EXEC GTBUTIL,SIZE=AUTO
 CATAL MAPSET=GTBDEMO
/*
 CLOSE SYSPCH,PUNCH
// DLBL IJSYSIN,'PUNCH.FILE'
// EXTENT. . . .
 ASSGN SYSIPT,DISK,. . . .
// EXEC PROC=GTBLIBS
* CATALOG PHYSICAL MAP
// OPTION CATAL
 INCLUDE
// EXEC LNKEDT
* CATALOG SYMBOLIC MAP
// EXEC LIBR,PARM='A S=user.source'
 CLOSE SYSIPT,SYSRDR
/&
```

Top of page

User's Guide Home
Administrator's Guide Home
BMS/TS Doc. Home

GTBUTIL COPY

Use the COPY function to make a duplicate of an existing member in the current BMS/TS system library.

> Caution: You must close the GT$FILE to use the COPY command.

COPY Format

Use the following format for the COPY function.

> COPY SET=*aaaaaaa*,MEMBER=*bbbbbbb*,NSET=*ccccccc*,NMEMBER=*ddddddd*
>
> **Note:** Type a comma to separate each parameter from any following parameter.

In order for the COPY function to work correctly, your JCL must include a DLBL or DD statement pointing to GT$FILE.

COPY Parameters

The following table describes the COPY parameters.

COPY Parameter	Description
MEMBER or **MAP**	**Required** –– Indicates the 1–8 character name of the map or member that you want to copy. **Note:** You can use a * wild card if necessary. For example, to delete all members beginning with GT, you would enter GT*.
NMEMBER or **NMAP**	**Required** –– Indicates the 1–8 character name of the duplicate map or member. **Note:** You can use a * wild card if necessary. For example, to delete all members beginning with GT, you would enter GT*.
NSET or **NTYPE**	**Required** –– Indicates name of the duplicate mapset or special set. Mapsets are 1–8 characters. Valid special set entries are *DEMO, *JCL,

	*MAPS, *OPID, *SYSTEM, *TABLE.
SET or **TYPE** or **MAPSET**	**Required** –– Indicates the name of the mapset or special set for the member you want to copy. Mapset names are 1–8 characters. Valid special set entries are *DEMO, *JCL, *MAPS, *OPID, *SYSTEM, *TABLE.
The following parameters are optional.	
NPASS	Indicates the password for the duplicate set or member.
PASS	Indicates the password of the set or member if you password protected the set/member. **Note:** If members of a set/type do not have the same password and you use a wildcard '*', you must supply the master password or BMS/TS displays the message Previous Function Abnormally Terminated. This message also displays if any update flags are set for a member.

Note: If you leave out a new name for a set/type, member/map, or password, BMS/TS uses the existing name or password.

COPY Example

The following sample COPY function duplicates a specific map in the BMS/TS system library.

```
COPY TYPE=GTBDEMO,MAP=MAP1,NTYPE=GTBDEMO,NMAP=MAP2
```

Top of page

User's Guide Home
Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

GTBUTIL CREATE

The CREATE function initializes the BMS/TS system library and creates the *ADMIN record.

> Caution: You must use the CREATE function before the online version of *ADMIN can operate.
>
> You must close the *xx*$FILE in order to use the CREATE command.

CREATE Format

Use the following format for the CREATE function.

> CREATE parameter(s)
>
> **Note:** Type a comma to separate each parameter from any following parameter.

In order for the CREATE function to work correctly, your JCL must include a DLBL or DD statement pointing to GT$FILE.

CREATE Parameters

The following table describes the unique CREATE parameters. All of these parameters are **optional**.

CREATE Parameter	Description
ALIGNED	Indicates whether the maps produced by BMS/TS should be aligned or unaligned. Valid entries are YES (aligned) or NO (unaligned). The default is **NO**.
MASTPASS	Allows you to define a 1–8 character master password that protects various system maintenance functions. This also overrides any member passwords.
NAME	Indicates the name you want to assign to your BMS/TS system library. The name can include up to 30 characters, but must begin and end with the same special character which cannot be used in the name itself. For example, if you wanted to call the library the Production Library, you would enter: +PRODUCTION LIBRARY+

Note: The remaining optional CREATE parameters are identical to those listed in the discussion of the ALTER function.

CREATE Examples

The following sample CREATE function initializes the *ADMIN record and adds a master password.

CREATE MASTPASS=*password*

The following sample CREATE function turns off the security signon feature.

CREATE SIGNON=*n*

The following sample CREATE function specifies the file containing the map and JCL libraries.

CREATE MAPLIB=*zt*,JCLLIB=*zt*

The following sample CREATE function specifies that the internal library containing the internal library members BMS/TS help screens and menus will be located in the ZT$FILE.

CREATE INTLIB=*zt*

Top of page

User's Guide Home
Administrator's Guide Home
BMS/TS Doc. Home

GTBUTIL DELETE

Use the DELETE function to remove an existing member from the current BMS/TS system library.

> Caution: You must close the *xx*$FILE in order to use this command.

DELETE Format

Use the following format for the DELETE function.

DELETE SET=*aaaaaaa*,MEMBER=*bbbbbbb*

Note: Type a comma to separate each parameter from any following parameter.

In order for the DELETE function to work correctly, your JCL must include a DLBL or DD statement pointing to GT$FILE.

DELETE Parameters

The following table describes the DELETE parameters.

DELETE Parameter	Description
MEMBER or **MAP**	**Required** –– Indicates the name of the map or member 1–8 characters that you want to delete. **Note:** You can use a * wild card if necessary. For example, to delete all members beginning with GT, you would enter GT*.
SET or **TYPE** or **MAPSET**	**Required** –– Indicates the name of the mapset or special set for the member you want to delete. Mapsets are 1–8 characters. Valid special set names are *DEMO, *JCL, *MAPS, *OPID, *SYSTEM, *TABLE.
PASS	Indicates the password of the set or member that you want to delete. This parameter is **required *only*** if you password–protected the member.

	Note: If members of a set/type do not have the same password and you use a wildcard '*', you must supply the master password or BMS/TS displays the message Previous Function Abnormally Terminated. This message also displays if any update flags are set for a member.

DELETE Example

The following sample DELETE function deletes a specific map in the BMS/TS system library.

```
DELETE SET=GTBDEMO,MEMBER=MAP1
```

Top of page

User's Guide Home
Administrator's Guide Home
BMS/TS Doc. Home

GTBUTIL DISKSEL

Use the DISKSEL function to back up specific members of the BMS/TS system library. DISKSEL can read a previous backup or history file and then let you select members from the current BMS/TS system library to be merged, producing a new history file.

> Caution: You must close the *xx*$FILE to updates to ensure that you get an accurate backup copy of the file.

DISKSEL Format

Use one of the following formats for the DISKSEL function:

DISKSEL MAPSET=*aaaaaaa*,DELETE=*xxx*,TYPE=*bbbb*

or

DISKSEL SET=**aaaaaaa*,MEMBER=*

Note: Type a comma to separate each parameter from any following parameter.

In order for the DISKSEL function to work correctly, your JCL must include a DLBL or DD statement pointing to GT$FILE.

BMS/TS sets the dataset control block information dynamically so you do not need to code the block. However, if you accidentally code the block, BMS/TS automatically overrides coded control blocks.

DISKSEL Parameters

The following table describes the DISKSEL parameters. All of these parameters are **optional**.

DISKSEL Parameter	Description
ALL	Indicates that you want to back up all members from the current BMS/TS system library to the backup tape. Valid entries are ALL (enable global backup) or blank (disable global backup).
DELETE	

	Indicates whether you want BMS/TS to move all selected mapsets in production status to the history file, then delete the mapsets from the BMS/TS system library. This lets you move a new set of maps into production status. Valid entries are YES or NO. **Note:** Use this parameter with extreme caution.
FROMDATE	Backup members created on/after the specified date. Enter the date in *ccyymmdd* format; this format does not change for European dates.
MAPSET	Indicates the name of the mapset you want to back up. You can select as many mapset names as desired. **Note:** You can use a * wild card if necessary. For example, to back up all mapsets beginning with GT, you would enter GT*.
MEMBER	Indicates the name of the member you want to back up. Required only if you enter a special set name. Keep in mind that members from the BMS/TS library replace the same named member in the history file.
REWIND	Indicates whether you want the tape rewound following the backup. Valid entries are YES or NO. The default is **NO**.
SET	Indicates the name of the special set you want to back up. Valid entries are *DEMO, *JCL, *MAPS, *OPID, *SYSTEM, *TABLE.
TODATE	Backup members created on/after the specified date. Enter the date in *ccyymmdd* format; this format does not change for European dates.
TYPE	Moves maps with a specific status code to the History file. Valid codes are TEST (test status maps) or PROD (production status maps).

DISKSEL Example

The following sample DISKSEL function moves all mapsets with specific names to the history file.

```
DISKSEL MAPSET=CUST*,MAPSET=GT*
```

DISKSEL JCL Samples

The following examples show JCL statements that use the DISKSEL function. Keep in mind that these are only examples; your JCL must be specific to the files and libraries used at your company.

DISKSEL Sample JCL (OS/390)

```
//BMSTS    JOB. . . .
//JOBCAT   DD   DSN=usercat,DISP=SHR
//GTBUTIL  EXEC PGM=GTBUTIL
//STEPLIB  DD   DSN=BMSTS.LOADLIB,DISP=SHR
//SYSLST   DD   SYSOUT=A
//GT$FILE  DD   DSN=GT.MASTER.FILE,DISP=SHR
//TAPIN    DD   DSN=xxxxxx, UNIT=TAPE,. . . . .
```

```
//TAPOUT   DD   DSN=xxxxxx, UNIT=TAPE,. . . . .
//SYSIN    DD   *
 DISKSEL SET=*JCL,MEMBER=SUBMIT
/*
```

Note: In the above example, //TAPIN DD DSN=xxxxxx, UNIT=TAPE,. . . indicates the tape unit address. If you want to create an output tape only, replace this statement with //TAPIN DD DUMMY. You can then use the output tape file with either the RESTORE or TAPESEL functions.

OS/390 can also back up to disk.

DISKSEL Sample JCL (VSE)

```
// JOB. . . .
// DLBL GT$FILE,'GT.MASTER.FILE',,VSAM
// TLBL TAPIN,. . . . .
// TLBL TAPOUT,. . . . .
// ASSGN SYS007,cuu
// ASSGN SYS008,cuu
// EXEC PROC=GTBLIBS
// EXEC GTBUTIL,SIZE=AUTO
 DISKSEL SET=*JCL,MEMBER=SUBMIT
/*
/&
```

Note: In the above example, ASSGN SYS007,cuu and ASSGN SYS008,cuu indicate the tape unit address. If you want to create an output tape only, replace cuu with IGN: for example, ASSIGN SYS007,IGN. You can then use the output tape file with either the RESTORE or TAPESEL functions.

Top of page

User's Guide Home
Administrator's Guide Home
BMS/TS Doc. Home

Contents:

Related topics:

- GTBUTIL Batch Utility

User's Guide Home
Administrator's Guide Home
BMS/TS Doc. Home

GTBUTIL DLIST

Use the DLIST function to produce a printed list of the entire BMS/TS system library.

DLIST Format

Use the following format for the DLIST function.

```
DLIST
```

DLIST Parameters

DLIST does not use any parameters. Simply include the DLIST command in your JCL.

Top of page

User's Guide Home
Administrator's Guide Home
BMS/TS Doc. Home

GTBUTIL MOVE

Use the MOVE function to transfer a member from one CICS library to another.

> Caution: You must close the *xx*$FILE in order to use this command.

MOVE Format

Use the following format for the MOVE function.

> MOVE FFILE=*aa*,TFILE=*bb*,SET=*xxxxxxx*,MEMBER=*yyyyyyy*
>
> **Note:** Type a comma to separate each parameter from any following parameter.

In order for the MOVE function to work correctly, your JCL must include a DLBL or DD statement pointing to GT$FILE.

MOVE Parameters

The following table describes the MOVE parameters.

MOVE Parameter	Description
FFILE	**Required** –– Indicates the 2–character prefix of the library from which you are transferring a members.
MEMBER or **MAP**	**Required** –– Indicates the 1–8 character name of the map or member that you want to move. **Note:** You can use a * wild card if necessary. For example, to move all members beginning with GT, you would enter GT*.
SET or **TYPE** or **MAPSET**	**Required** –– Indicates the 1–8 character name of the mapset or special set for the member you want to move. Valid special set names are *DEMO, *JCL, *OPID, *SYSTEM, *TABLE.

	Note: If you are moving a member from a special set other than *MAPS, you must use the SET parameter.
TFILE	**Required** –– Indicates the 2–character prefix of the library to which you are transferring a members.
The following parameters are optional.	
DELETE	Indicates whether BMS/TS should delete the mapset from the sending library after the transfer. Valid entries are YES or NO. The default is **NO**.
REPLACE	Indicates whether BMS/TS should replace duplicate mapsets in the receiving library with the transferred mapsets. Valid entries are YES or NO. The default is **NO**.

MOVE Example

The following sample MOVE function transfers all maps in the GTBDEMO mapset of the GT library to the AT library.

```
MOVE FFILE=GT,TFILE=AT,SET=GTBDEMO,MEMBER=*
```

MOVE JCL Samples

The following examples show JCL statements that use the MOVE function. Keep in mind that these are only examples; your JCL must be specific to the files and libraries used at your company.

MOVE Sample JCL (OS/390)

```
//BMSTS   JOB. . . .
//JOBCAT  DD   DSN=usercat,DISP=SHR
//GTBUTIL EXEC PGM=GTBUTIL
//STEPLIB DD   DSN=BMSTS.LOADLIB,DISP=SHR
//SYSLST  DD   SYSOUT=A
//GT$FILE DD   DSN=GT.MASTER.FILE,DISP=SHR
//AT$FILE DD   DSN=AT.MASTER.FILE,DISP=SHR
//SYSIN   DD   *
 MOVE FFILE=GT,TFILE=AT,SET=GTBDEMO,MEMBER=*
/*
```

MOVE Sample JCL (VSE and VSE/SP)

```
// JOB. . . .
// DLBL GT$FILE,'GT.MASTER.FILE',,VSAM
// DLBL GT$FILE,'AT.MASTER.FILE',,VSAM
// EXEC PROC=GTBLIBS
// EXEC GTBUTIL,SIZE=AUTO
 MOVE FFILE=GT,TFILE=AT,SET=GTBDEMO,MEMBER=*
/*
```

```
/&
```

Top of page

User's Guide Home
Administrator's Guide Home
BMS/TS Doc. Home

GTBUTIL PASSWRD

Use the PASSWRD function to add or change the password of a member in the current BMS/TS system library.

> Caution: You must close the *xx*$FILE in order to use this command.

PASSWRD Format

Use the following format for the PASSWRD function.

> PASSWRD SET=*aaaaaaa*,MEMBER=*bbbbbbb*
>
> **Note:** Type a comma to separate each parameter from any following parameter.

In order for the PASSWRD function to work correctly, your JCL must include a DLBL or DD statement pointing to GT$FILE.

PASSWRD Parameters

The following table describes the PASSWRD parameters.

PASSWRD Parameters	Description
MEMBER or **MAP**	**Required** Indicates the 1–8 character name of the map or member for which you want to add or update a password. **Note:** You can use a * wild card if necessary. For example, to add or update passwords for all members beginning with GT, you would enter GT*.
SET or **TYPE** or **MAPSET**	**Required** Indicates the 1–8 character name of the mapset or special set for the member for which you want to add or update a password. Valid special set names are *DEMO, *JCL, *MAPS, *OPID, *SYSTEM, *TABLE.

The following parameters are optional.	
NPASS	Indicates the new password for the set or member.
PASS	Indicates the password of the set or member that you want to delete if you password protected the set/member. **Note:** If members of a set/type do not have the same password and you use a wildcard '*', you must supply the master password or BMS/TS displays the message Previous Function Abnormally Terminated. This message also displays if any update flags are set for a member.

PASSWRD Example

The following sample PASSWRD function changes the password for a map in the BMS/TS system library.

```
PASSWRD SET=GTBDEMO,MEMBER=MAP1,PASS=1234,NPASS=4321
```

Top of page

User's Guide Home
Administrator's Guide Home
BMS/TS Doc. Home

GTBUTIL PRINT

Use the PRINT function to print a display of a requested map, along with a list of all fields defined for the map. The report can also include all system defaults.

PRINT Format

Use one of the following formats for the PRINT function:

PRINT MAPSET=*aaaaaaa*,MAP=*bbbbbbb*
or
PRINT SET=**aaaaaaa*,MEMBER=*bbbbbbb*

Note: Type a comma to separate each parameter from any following parameter.

In order for the PRINT function to work correctly, your JCL must include a DLBL or DD statement pointing to GT$FILE.

PRINT Parameters

The following table describes the PRINT parameters. All of these parameters are **optional**.

PRINT Parameters	Description
***ADMIN**	Allows you to print the contents of the *ADMIN record, which contains the BMS/TS system defaults. If the *ADMIN record has been protected with the Master Password, you must also include the MASTPASS=*xxxxxxxx* parameter, where *xxxxxxxx* is the Master Password.
DSECT	Indicates whether to print the symbolic map. Valid entries are YES or NO. The default is **YES**.
FILLER	Indicates whether to print a special pattern wherever a name field display only exists on the physical map. Valid entries are YES or NO. The default is **YES**.

MAP	Indicates the name of the map whose display and defined fields you want to print. Required only if you enter a mapset name. Valid entries are: • * –– Print all members in a mapset, • *aaaaaaa* –– Where *aaaaaaa* represents the name of the map or member you want to print.
MAPPRT	Indicates whether to print the physical map. Valid entries are YES or NO. The default is **YES**.
MAPSET	Indicates the name of the mapset whose display and defined fields you want to print. You can select as many mapset names as desired. **Note:** You can use a * wild card if necessary. For example, to print the display and defined fields of all mapsets beginning with GT, you would enter GT*.
MASTPASS	Indicates the BMS/TS master password. This parameter is required if you are printing the *ADMIN record and it has been protected with the master password. There is no default.
MEMBER	Indicates the name of the map or member whose display and defined fields you want to print. Required only if you enter a special set name. **Note:** You can use a * wild card if necessary. For example, to print the display and defined fields of all members beginning with GT, you would enter GT*.
SET	Indicates the name of the special set whose maps you want to print. Valid entries are *JCL, *MAPS, *OPID, *SYSTEM, *TABLE.
SF	Indicates whether to print a single special character wherever an attribute byte exists on the physical map. Valid entries are: • NONE –– The attribute byte does not print, • NO –– The attribute byte does not print, • *a* –– Where "*a*" represents the character you want to print in place of all attribute bytes. The default is @.
SPACE	Indicates whether to print a single special character wherever a true space exists on the physical map. This shows you which bytes of the map are true spaces hexadecimal 40 instead of nulls hexadecimal/binary 00. Valid entries are: • NO –– Special character does not print. • *a* –– Where "*a*" represents the character you want to print in place of all true spaces. The default is "**?**".
VALID	Indicates whether to print any input rules associated with the map. Valid entries are YES or NO. The default is **NO**.

PRINT Examples

The following sample PRINT function prints a specific map. The report includes special characters in place of both attribute bytes and true spaces on the physical map.

```
PRINT MAPSET=GTBDBST,MAP=MAP1,SF=#,SPACE=&,FILLER=NO
```

The following sample PRINT function prints the *ADMIN record. Notice that the MASTPASS parameter is required if *ADMIN has been protected with a master password.

```
PRINT *ADMIN,MASTPASS=xxxxxxxx
```

Top of page

User's Guide Home
Administrator's Guide Home
BMS/TS Doc. Home

GTBUTIL PROD

Use the PROD function to change the status of a member in the BMS/TS system library from test to production.

> Caution: You must close the *xx*$FILE in order to use this command.

PROD Format

Use the following format for the PROD function.

PROD MAPSET=*aaaaaaa*,MAP=*bbbbbbb*

Note: Type a comma to separate each parameter from any following parameter.

In order for the PROD function to work correctly, your JCL must include a DLBL or DD statement pointing to GT$FILE.

PROD Parameters

The following table describes the PROD parameters.

PROD Parameters	Description
MAP	**Required** –– Indicates the name of the map whose status you want to change. **Note:** You can use a * wild card if necessary. For example, to delete all members beginning with GT, you would enter GT*.
MAPSET	**Required** –– Indicates the name of the mapset for the member whose status you want to change. Mapset names are 1–8 characters.
The following parameter is optional.	
PASS	Indicates the password of the set or member if you password protected the set/member. **Note:** If members of a set/type do not have the same password and you use a wildcard '*', you must supply the master password or

	BMS/TS displays the message Previous Function Abnormally Terminated. This message also displays if any update flags are set for a member.

PROD Example

The following sample PROD function modifies a specific map from test to production status.

```
PROD MAPSET=GTBDEMO,MEMBER=MAP1
```

Top of page

User's Guide Home
Administrator's Guide Home
BMS/TS Doc. Home

GTBUTIL PUNCH

Use the PUNCH function to move a specific table into the BMS/TS system library.

PUNCH Format

Use the following format for the PUNCH function.

PUNCH SET=*TABLE,MEMBER=*aaaaaaa*

Note: Type a comma to separate each parameter from any following parameter.

In order for the PUNCH function to work correctly, your JCL must include a DLBL or DD statement pointing to GT$FILE.

PUNCH Parameters

The following table describes the PUNCH parameters.

PUNCH Parameters	Description
SET	**Required** –– Indicates the name of the special set for the table you want to move into the BMS/TS system library. Always ***TABLE**.
MEMBER	**Required** –– Indicates the name of the table you want to move into the BMS/TS system library.

PUNCH Example

The following sample PUNCH function moves a specific table to the BMS/TS system library.

PUNCH SET=*TABLE,MEMBER=TABLE1

PUNCH JCL Samples

The following examples show JCL statements that use the PUNCH function. Keep in mind that these are only examples; your JCL must be specific to the files and libraries used at your company.

PUNCH Sample JCL (OS/390)

```
//BMSTS    JOB. . . .
//JOBCAT   DD   DSN=usercat,DISP=SHR
//GTBUTIL  EXEC PGM=GTBUTIL
//STEPLIB  DD   DSN=BMSTS.LOADLIB,DISP=SHR
//SYSLST   DD   SYSOUT=A
//GT$FILE  DD   DSN=GT.MASTER.FILE,DISP=SHR
//GT$PCHP  DD   DSN=&&PCHP,UNIT=SYSDA,SPACE=CYL,1,
                DISP=,PASS,DCB=BLKSIZE=80
//SYSIN    DD   *
PUNCH SET=*TABLE,MEMBER=TABLE1
/*
//IEBUPDTE EXEC PGM=IEBUPDTE
//SYSPRINT DD   SYSOUT=A
//SYSUT1   DD   DSN=users.objlib,DISP=SHR
//SYSUT2   DD   DSN=users.objlib,DISP=SHR
//SYSIN    DD   DSN=*.GTBUTIL.GT$PCHP,
//              DISP=OLD,DELETE,DELETE
//LINK     EXEC PGM=IEWL,PARM='LIST,NCAL,LET, XREF
//SYSPRINT DD   SYSOUT=A
//SYSLMOD  DD   DSN=BMSTS.LOADLIB,DISP=SHR
//SYSUT1   DD   UNIT=SYSDA,SPACE 1024, 100,10
//SYSLIB   DD   DSN=users.objlib,DISP=SHR
//SYSLIN   DD   *
 INCLUDE SYSLIBmmmmmmmm
 NAME mmmmmmmmR
/*
```

PUNCH Sample JCL (VSE and VSE/SP)

```
// JOB. . . .
// DLBL GT$FILE,'GT.MASTER.FILE',,VSAM
// DLBL IJSYSPH,'PUNCH.FILE'
// EXTENT. . . .
 ASSGN SYSPCH,DISK,. . . .
// EXEC PROC=GTBLIBS
// EXEC GTBUTIL,SIZE=AUTO
 PUNCH SET=*TABLE,MEMBER=TABLE1
/*
 CLOSE SYSPCH,PUNCH
// DLBL IJSYSIN,'PUNCH.FILE'
// EXTENT. . . .
 ASSGN SYSIPT,DISK,. . . .
// EXEC PROC=GTBLIBS
* CATALOG PHYSICAL MAP
// OPTION CATAL
 PHASE mmmmmmmm,*
 INCLUDE
// EXEC LNKEDT
 CLOSE SYSIPT,SYSRDR
/&
```

Top of page

User's Guide Home
Administrator's Guide Home
BMS/TS Doc. Home

GTBUTIL RENAME

Use the RENAME function to change the name of a member in the current BMS/TS system library.

> Caution: You must close the *xx*$FILE in order to use this command.

RENAME Format

Use the following format for the RENAME function.

RENAME SET=*aaaaaaa*,MEMBER=*bbbbbbb*,NSET=*xxxxxxx*,NMEMBER=*yyyyyyy*

Note: Type a comma to separate each parameter from any following parameter.

In order for the RENAME function to work correctly, your JCL must include a DLBL or DD statement pointing to GT$FILE.

RENAME Parameters

The following table describes the RENAME parameters.

RENAME Parameters	Description
MEMBER or **MAP**	**Required** –– Indicates the name of the map or member that you want to rename. **Note:** You can use a * wild card if necessary. For example, to rename all members beginning with GT, you would enter GT*.
NMEMBER or **NMAP** *	**Required** –– Indicates the new name of the map or member.
NSET or **TYPE** *	**Required** –– Indicates the new name of the mapset or special set for the member you want to rename. Mapsets are 1–8 characters. Valid special set names are *DEMO, *JCL, *MAPS, *OPID, *SYSTEM, *TABLE.

SET or **TYPE** or **MAPSET**	**Required** –– Indicates the name of the mapset or special set for the member you want to rename. Mapsets are 1–8 characters. Valid special set names are *DEMO, *JCL, *MAPS, *OPID, *SYSTEM, *TABLE.
The following parameters are optional.	
NPASS *	Indicates the new password for the set or member.
PASS	Indicates the password of the set or member that you want to rename if you password protected the set/member. **Note:** If members of a set/type do not have the same password and you use a wildcard '*', you must supply the master password or BMS/TS displays the message Previous Function Abnormally Terminated. This message also displays if any update flags are set for a member.
* Note: If you do not specify a new name for a set/type, member/map, or password, BMS/TS uses the existing name or password.	

RENAME Example

The following sample RENAME function changes the name of a specific map in the BMS/TS system library.

```
RENAME SET=GTBDEMO,MEMBER=MAP1,NSET=GTBDEMO,NMEMBER=MAP2
```

Top of page

User's Guide Home
Administrator's Guide Home
BMS/TS Doc. Home

GTBUTIL RESET

Use the RESET function to change the update flag of a set and/or member in the current BMS/TS system library so that you can edit the set/member.

> **Note:** You only need to use this function when CICS abnormally terminates, leaving a member in an "update" status.
>
> Caution: You must close the *xx*$FILE in order to use this command.

RESET Format

Use the following format for the RESET function.

> RESET SET=*aaaaaaa*,MEMBER=*bbbbbbb*
>
> **Note:** Type a comma to separate each parameter from any following parameter.

In order for the RESET function to work correctly, your JCL must include a DLBL or DD statement pointing to GT$FILE.

RESET Parameters

The following table describes the RESET parameters.

RESET Parameters	Description
MEMBER or **MAP**	**Required** –– Indicates the name of the map or member whose update flag you want to reset. **Note:** You can use a * wild card if necessary. For example, to reset the update flags for all members beginning with GT, you would enter GT*.
SET or **TYPE** or **MAPSET**	**Required** –– Indicates the name of the mapset or special set for the member whose update flag you want to reset. Mapsets are 1–8 characters. Valid special set names are *DEMO, *JCL, *MAPS, *OPID, *SYSTEM, *TABLE.
The following parameter is optional.	
PASS	Indicates the password of the set or member whose update flag you want to reset if you password protected the set/member.

	Note: If members of a set/type do not have the same password and you use a wildcard '*', you must supply the master password or BMS/TS displays the message Previous Function Abnormally Terminated. This message also displays if any update flags are set for a member.

RESET Example

The following sample RESET function resets the update flag for a specific map in the BMS/TS system library.

```
RESET SET=GTBDEMO,MEMBER=MAP1
```

Top of page

User's Guide Home
Administrator's Guide Home
BMS/TS Doc. Home

GTBUTIL RESTORE

Use the RESTORE function to recover a previous BMS/TS system library backup, including the password and expiration date of the software.

> Caution: The BMS/TS system library must have an empty status before you perform this function.
>
> You must close the *xx*$FILE in order to use this command.

RESTORE Format

Use the following format for the RESTORE function.

> RESTORE LIST=xxx,FROMDATE=*yyyymmdd*,TODATE=*yyyymmdd*
>
> **Note:** In order for the RESTORE function to work correctly, your JCL must include a DLBL or DD statement pointing to GT$FILE.

Separate multiple parameters with a comma.

RESTORE Parameters

The following table describes the RESTORE parameters. All of these parameters are **optional**.

RESTORE Parameters	Description
FROMDATE	Restore members created on/after the specified date. Enter the date in *ccyymmdd* format; this format does not change for European dates.
LIST	Produces a list of the members written from the backup dataset. Valid entries are YES or NO.
TODATE	Restore members created on/before the specified date. Enter the date in *ccyymmdd* format; this format does not change for European dates.

RESTORE Example

The following sample RESTORE function restores a previous BMS/TS system library backup, and instructs BMS/TS to include a list of members written to the tape.

RESTORE JCL Samples

The following examples show JCL statements that use the RESTORE function. Keep in mind that these are only examples; your JCL must be specific to the files and libraries used at your company.

RESTORE Sample JCL (OS/390)

```
//BMSTS    JOB. . . .
//JOBCAT   DD   DSN=usercat,DISP=SHR
//GTBUTIL  EXEC PGM=GTBUTIL
//STEPLIB  DD   DSN=BMSTS.LOADLIB,DISP=SHR
//SYSLST   DD   SYSOUT=A
//GT$FILE  DD   DSN=GT.MASTER.FILE,DISP=SHR
//TAPIN    DD   DSN=xxxxxx, UNIT=TAPE,. . . . .
//SYSIN    DD   *
 RESTORE
/*
```

RESTORE Sample JCL (VSE)

```
// JOB. . . .
// DLBL GT$FILE,'GT.MASTER.FILE',,VSAM
// TLBL TAPIN,. . . . .
// ASSGN SYS007,cuu
// EXEC PROC=GTBLIBS
// EXEC GTBUTIL,SIZE=AUTO
 RESTORE LIST=YES
/*
/&
```

Top of page

User's Guide Home
Administrator's Guide Home
BMS/TS Doc. Home

GTBUTIL SETFILE

Use the SETFILE function to redirect all input and output requests to a specific CICS library. This enables you to access multiple CICS libraries.

SETFILE Format

Use the following format for the SETFILE function.

```
SETFILE PREFIX=aa
```

Note: In order for the SETFILE function to work correctly, your JCL must include a DLBL or DD statement pointing to GT$FILE.

SETFILE Parameters

The following table describes the SETFILE parameter.

SETFILE Parameter	Description
PREFIX	**Required** –– Indicates the 2–character prefix of the library to which you want to redirect input and output. **Note:** BMS/TS uses the library you specify here until you issue another SETFILE function.

SETFILE Example

The following sample SETFILE function redirects all input and output to the AT library.

```
SETFILE PREFIX=AT
```

SETFILE JCL Samples

The following examples show JCL statements that use the SETFILE function. Keep in mind that these are only examples; your JCL must be specific to the files and libraries used at your company.

SETFILE Sample JCL (OS/390)

```
//BMSTS    JOB. . . .
//JOBCAT   DD   DSN=usercat,DISP=SHR
//GTBUTIL  EXEC PGM=GTBUTIL
//STEPLIB  DD   DSN=BMSTS.LOADLIB,DISP=SHR
//SYSLST   DD   SYSOUT=A
//GT$FILE  DD   DSN=GT.MASTER.FILE,DISP=SHR
//AT$FILE  DD   DSN=AT.MASTER.FILE,DISP=SHR
//SYSIN    DD   *
 DLIST
 SETFILE PREFIX=AT
 DLIST
/*
```

SETFILE Sample JCL (VSE and VSE/SP)

```
// JOB. . . .
// DLBL GT$FILE,'GT.MASTER.FILE',,VSAM
// DLBL AT$FILE,'AT.MASTER.FILE',,VSAM
// EXEC PROC=GTBLIBS
// EXEC GTBUTIL,SIZE=AUTO
 DLIST
 SETFILE PREFIX=AT
 DLIST
/*
/&
```

Top of page

User's Guide Home
Administrator's Guide Home
BMS/TS Doc. Home

GTBUTIL TAPESEL

Use the TAPESEL function to read a previous backup or history file and copy selected members back into the current BMS/TS system library.

> Caution: You must close the *xx*$FILE in CICS in order to use the TAPESEL function.

Separate multiple parameters with a comma.

You can only use one TAPESEL in a single execution of GTBUTIL.

TAPESEL Format

Use one of the following formats for the TAPESEL function:

TAPESEL MAPSET=*aaaaaaa*,REPLACE=*xxx*
or
TAPESEL ALL,REPLACE=*xxx*
or
TAPESEL SET=**aaaaaaa*,MEMBER=*aaaaaaa*

Note: Type a comma to separate each parameter from any following parameter.

In order for the TAPESEL function to work correctly, your JCL must include a DLBL or DD statement pointing to GT$FILE.

TAPESEL Parameters

The following table describes the TAPESEL parameters. All of these parameters are **optional**.

TAPESEL Parameters	Description
ALL	Indicates that all members from the backup tape should merge into the current BMS/TS system library. Valid entries are ALL (enable global restore) or blank (disable global restore). **Note:** If duplicate members appear in the library, BMS/TS does not overwrite them.
FROMDATE	

	Copy members created on/after the specified date. Enter the date in *ccyymmdd* format; this format does not change for European dates.
MAPSET	Indicates the name of the mapset you want to restore. You can select as many mapset names as desired. **Note:** You can use a * wild card if necessary. For example, to restore all mapsets beginning with GT, you would enter GT*.
MEMBER	Indicates the name of the member you want to restore. Required only if you enter a special set name. Keep in mind that BMS/TS does not overwrite a member if a duplicate appears in the current library.
REPLACE	Indicates whether you want members from the backup tape to replace members in the BMS/TS system library with the same name. Valid entries are YES or NO. The default is **NO**.
SET	Indicates the name of the special set you want to restore. Valid entries are *DEMO, *JCL, *MAPS, *OPID, *SYSTEM, *TABLE.
TODATE	Copy members created on/before the specified date. Enter the date in *ccyymmdd* format; this format does not change for European dates.

TAPESEL Example

The following sample TAPESEL function moves all maps on the backup tape to the BMS/TS system library.

```
TAPESEL ALL
```

TAPESEL JCL Samples

The following examples show JCL statements that use the TAPESEL function. Keep in mind that these are only examples; your JCL must be specific to the files and libraries used at your company.

TAPESEL Sample JCL (OS/390)

```
//BMSTS    JOB. . . .
//JOBCAT   DD   DSN=usercat,DISP=SHR
//GTBUTIL  EXEC PGM=GTBUTIL
//STEPLIB  DD   DSN=BMSTS.LOADLIB,DISP=SHR
//SYSLST   DD   SYSOUT=A
//GT$FILE  DD   DSN=GT.MASTER.FILE,DISP=SHR
//TAPIN    DD   DSN=xxxxxx, UNIT=TAPE,. . . . .
//SYSIN    DD *
 TAPESEL ALL
/*
```

TAPESEL Sample JCL (VSE)

```
// JOB. . . .
// DLBL GT$FILE,'GT.MASTER.FILE',,VSAM
// TLBL TAPIN,. . . . .
// ASSGN SYS007,cuu
// EXEC PROC=GTBLIBS
// EXEC GTBUTIL,SIZE=AUTO
 TAPESEL ALL
/*
/&
```

Top of page

User's Guide Home
Administrator's Guide Home
BMS/TS Doc. Home

Appendix C, BMS/TS Transaction Codes

The table below lists and describes BMS/TS transaction codes.

Code	Description
GTB/GTBA	Displays the main system menu. Using the GTB transaction, you can add, update, or display members on the product library. GTBA is the same as GTB, except that it supports alternate screen sizes.
GTBH/GTBI	Activates and deactivates the product by loading and unloading (GTBH Deload) the Transaction Monitor table. It is the transaction used during user help screen displays, and Application Interface processing. GTBI is the same as GTBH, except that it supports alternate screen sizes.
GTBS	Initiates the Installation Status Review screen. The status screen displays information about the installation; it is used for troubleshooting as well as during the install checkout process.
GTC/GTCA	HFS Command Line Processor. GTCA is the same as GTC, except that it supports alternate screen sizes.
GTIB	Password Administration function.
GTLB	NLS (National Language Support) administration.
GTP	Printer configuration.
GTRV	FTP Listener and Server.
GTTT	3270 Bridge Test Tool.

Top of page

User's Guide Home
Administrator's Guide Home
ASSIST/TS Doc. Home

Contents:

Appendix D, Support for Multi–Byte Character Sets

DBCS (Double–Byte Character Set) or MBCS (Multi–Byte Character Set) refer to character sets that require more than one character to compose a display character. Latin based languages use SBCS (Single byte Character Set) to define characters.

BMS/TS provides support for multi–byte character set (MBCS) languages such as Chinese, Japanese, Thai, etc. during map design and map creation. You can migrate BMS screens which were not previously designed for MBCS by simply recreating them during the SUBMIT function. This adds the BMS “SOSI” (Shift–Out/Shift–In) support to the map and DSECT copybooks.

SOSI refers to the method of using two special character values to allow switching from SBCS to DBCS or MBCS languages. Languages such as Chinese and Japanese rely on SOSI. Languages such as Thai, which is a true multi–byte language, rely on rules to define the order of characters and do not use SOSI.

For more refined or detailed requirements, you can design screens to include MBCS characters as screen constants and/or the individual control of each screen field to determine whether MBCS characters are allowed.

> Caution: In order to develop maps using MBCS, you must have the Program Symbol turned **on** for your terminal. See your CICS Systems Programmer for assistance.

Basic Mapping Support

IBM has implemented basic Mapping Support (BMS) for MBCS support using the following procedures:

- **SOSI** – Allows the application to send/receive data that may or may not contain MBCS characters. The determining factor is the presence of the SO character (x’0E’) preceding MBCS characters and the SI character (x’0F’) indicating that any data following will be SBCS (Single–Byte Character Set). For additional information, refer to the appropriate CICS reference manuals.

- **Programmed Symbols** – Acts as an indicator that all data within the field is treated as MBCS data. Currently, the allowed values are x’00’ or x’F8’. Programmed symbol ‘8’ (x’F8’) indicates the field contents are to be treated as MBCS data. For compatibility with ASCII versions of CICS, the ASCII ‘8’ (x’38’) is also supported and provides the same functionality. For additional

information, refer to the appropriate CICS reference manuals.

BMS/TS MBCS Considerations

You may need to consider BMS/TS support for MBCS when you:

- Design a map,
- Redesign a map,
- Submit a map,
- Convert/import an existing map.

MBCS Support in BMS/TS Map Design

During the map options display, you can specify SOSI=YES to allow MBCS data to be entered while designing the screen. Then, design the map. For MBCS fields, use the Field Mark key to represent any MBCS initial value. During field definition, you will be prompted to enter the initial value desired. In this way, BMS/TS can help the designer in defining the appropriate field size and attributes.

After you press Enter from the design screen, BMS/TS proceeds to field definitions. Aside from the typical field data, you may choose to enter the hex value F8 into the Programmed Symbol field, or type Y in the SOSI option field. Either way, the user will be presented with a panel to enter any initial value desired. BMS/TS will display a field of the appropriate size and attributes in which to key this data. This helps ensure the data and length attributes are properly set.

As before, when all fields have been defined, BMS/TS will present a verification display of the newly created/updated map (with the MBCS data embedded).

MBCS Support in BMS/TS Map Redesign

Screen redesign works the same as for non–MBCS maps. During the initial screen representations, MBCS data is signified using the Field Mark character. If you wish to change and examine initial MBCS data, use the Change field character to mark the field you wish to examine.

MBCS Support in BMS/TS Map Submission Options

If the map was designed with SOSI, you do not need to select options at submit time. If the map was not designed for SOSI and you want to do a quick conversion, select the SOSI option to enable SOSI for all Named fields.

During the map build (submit) process online or batch, you can specify SOSI=Yes to indicate that the map being submitted will be created for MBCS using SOSI. This allows the application to request a field be enabled for SOSI and send/receive MBCS data within these selected fields.

MBCS Support in BMS/TS Map Conversion (Import)

The batch import utility (GTBCONV) supports maps using SOSI and/or PS=8 MBCS support.

Top of page

Getting Technical Support

This document explains how to get technical assistance if you have questions about BMS/TS.

Product Documentation

BMS/TS is supported by online help. The help system is the first place you should look when you need information about the product.

If the online help does not answer your question, refer to the appropriate sections of the product documentation for more detailed information.

You can also **search the product support database** on the GT Web site at <http://www.gtsoftware.com>.

Technical Support Staff

If you cannot resolve the problem using the product documentation, the GT Software Technical Support staff is available for telephone consultation. Before you call, have the following information available:

- BMS/TS release number,
- Transaction Server release number,
- The number and text of the system message (if any),
- Complete description of the problem.

In order to resolve a problem, it is important to tell the Technical Support representative exactly what was being done when the problem occurred. Details are important. It may be helpful to make notes about what happened before you call.

You can also try to re–create the problem, and tell the Technical Support representative whether you got the same results.

Contacting GT Software

GT Software Technical Support representatives are available Monday through Friday between the hours of 8:30 AM and 5:15 PM Eastern time. After business hours, you can leave a message. Your call will be returned the next business day.

To call GT Software, dial 404–253–1300.

To send a facsimile (fax), dial 404–253–1314.

Send Internet e–mail to sales@gtsoftware.com or support@gtsoftware.com.

Visit the GT Software Web site at <http://www.gtsoftware.com>.

Top of page

BMS/TS Doc. Home
User's Guide Home
Administrator's Guide Home

Contents:

Administrator's Guide Index

Top of page

Administrator's Guide Home
BMS/TS Doc. Home